



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A
BIOMECHANIKY**

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

**NÁVRH ŘÍDICÍHO ALGORITMU PRO POLOHOVACÍ
PLATFORMU**

DESIGN OF CONTROL SYSTEM FOR MANIPULATING PLATFORM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Matúš Remiš

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Zdeněk Hadaš, Ph.D.

BRNO 2022

Zadání diplomové práce

Ústav: Ústav mechaniky těles, mechatroniky a biomechaniky
Student: **Bc. Matúš Remiš**
Studijní program: Mechatronika
Studijní obor: bez specializace
Vedoucí práce: **doc. Ing. Zdeněk Hadaš, Ph.D.**
Akademický rok: 2021/22

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Návrh řídicího algoritmu pro polohovací platformu

Stručná charakteristika problematiky úkolu:

Polohovací jednotky jsou jedním z klíčových zařízení v automatizaci i jednoúčelových aplikacích. Srdcem každé automatizační procedury je manipulátor, ať už je potřeba pohybovat senzorem, detektorem nebo vyráběným dílem. Často však není potřeba zařízení konstruovat od základu, ale je možné přizpůsobit hotová řešení. Například požadavek na větší přesnost může být v jistých případech nahrazen zvýšenou nostností a rozsahem pohybu.

Cíle diplomové práce:

- řešerše polohovacích konstrukcí a jejich řízení
- návrh pohonné jednotky
- návrh senzorických systémů
- vytvoření uživatelského prostředí platformy

Seznam doporučené literatury:

SCHIEHLEN, W., 1997. Multibody System Dynamics: Roots and Perspectives. Multibody System Dynamics [online]. 1(2), 149–188.

MĚŘIČKA, Jiří a Zdeněk ZOUBEK, 1973. Obecná teorie elektrického stroje. B.m.: SNTL.

WILSON, Peter. The Circuit Designer's Companion. Oxford: Elsevier Science & Technology, 2017. ISBN 0081017642.

KÖHL, Susanne a Dirk JEGMINAT. How to Do Hardware-in-the-Loop Simulation Right. 2005. ISSN 0148-7191. Dostupné z: doi:10.4271/2005-01-1657

LOW, Kay-soon a M.-T KECK. Advanced precision linear stage for industrial automation applications. IEEE transactions on instrumentation and measurement. IEEE, 2003, 52(3), 785-789.

HVILSHØJ, Mads a Simon BØGH. "Little Helper" — An Autonomous Industrial Mobile Manipulator Concept. International journal of advanced robotic systems [online]. London, England: SAGE Publications, 2011, 8(2), 15.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2021/22

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Táto diplomová práca je súčasťou projektu, ktorého riešenie bolo pridelené dvojici študentov. Cieľom projektu je návrh a realizácia bezdrôtového riadiaceho systému pre polohovaciu platformu. Navrhnutý systém umožňuje napájanie polohovacej platformy pomocou batérie a bezdrôtové riadenie z PC. Jednotlivé časti projektu sa zaoberajú koncepciou funkcionality riadiaceho systému, návrhom a výrobou prototypu riadiacej jednotky, návrhom jej firmwaru a softwaru užívateľského prostredia.

Časti projektu, ktorými sa zaoberá táto práca, sú návrh firmwaru riadiacej jednotky a užívateľského softwaru. V rámci tejto práce bola taktiež zostavená testovacia pohonná jednotka, ktorá po prepojení s riadiacou jednotkou slúži na verifikáciu navrhovaných funkcionality systému.

Navrhnutý riadiaci systém bol úspešne testovaný. Výstupom práce je preto riadiaci systém, ktorý je možné spolu s riadiacou jednotkou v rámci projektu ďalej rozvíjať.

Kľúčové slová

polohovacia platforma, krokový motor, riadiaci firmware, bezdrôtové riadenie, C, C#

Abstract

This thesis is a part of a project, in which two students participated. The aim of the project is the design and implementation of a wireless control system for the manipulation XY platform. This control system allows the manipulation platform to be battery powered and controlled wirelessly from PC. The project consists of the control system concept design, design of an electronic control unit prototype, firmware design and user control software.

The parts of the project that are the subject of this thesis are firmware and software design. Within this work, a test drive unit was also created, which after connection with the control unit serves to verify the functionalities of the control system.

The proposed control system was successfully tested. The result of this work is therefore firmware and software, which can be further developed within the project.

Keywords

manipulation platform, stepper motor, control firmware, wireless control, C, C#

Bibliografická citácia:

REMIŠ, Matúš. *Návrh řídicího algoritmu pro polohovací platformu* [online]. Brno, 2022 [cit. 2022-05-18]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/139926>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Zdeněk Hadaš.

Čestné prehlásenie

„Prehlasujem, že som túto diplomovú prácu na tému Návrh řídicího algoritmu pro polohovací platformu vypracoval samostatne pod vedením vedúceho práce doc. Ing. Zdeňka Hadaše, Ph.D. Vychádzal som pritom zo svojich znalostí, odborných konzultácií a zdrojov uvedených na konci tejto práce.“

V Brne dňa: **18.5.2022**

Bc. Matúš Remiš

Pod'akovanie

„Ďakujem vedúcemu práce doc. Ing. Zdeňkovi Hadašovi, Ph.D za priateľský prístup a cenné rady pri spracovaní tejto diplomovej práce. Ďalej ďakujem kolegovi Adamovi Ondryášovi za výbornú spoluprácu a pomoc pri riešení problémov. Na záver chcem poďakovať svojej rodine, priateľke a kamarátom za podporu počas celého štúdia.“

Obsah

1	Úvod.....	10
2	Definícia problémovej situácie a stanovenie cieľov práce.....	11
2.1	Členovia tímu a rozdelenie úloh	12
3	Polohovacie platformy a ich riadenie.....	13
3.1	Pojazdové mechanizmy osí.....	13
3.1.1	Lineárny motor	14
3.1.2	Piezomotor.....	14
3.1.3	Kombinácia rotačného motora a ozubeného hrebeňa.....	15
3.1.4	Kombinácia rotačného motora a ozubeného remeňa.....	15
3.1.5	Kombinácia rotačného motora a pohybovej skrutky	16
3.2	Analýza komerčne dostupných polohovacích platforiem.....	17
3.2.1	Výrobcovia polohovacích platforiem	17
3.2.2	Typy využívaných pohonov	17
3.2.3	Komerčne dostupné polohovacie platformy	18
3.2.4	Vhodný typ pohonu testovacej jednotky	19
3.3	Krokové motory	20
3.3.1	Typy krokových motorov	20
3.3.2	Hybridný krokový motor	21
3.3.3	Vinutia krokových motorov.....	22
3.4	Radiče krokových motorov	23
3.4.1	Základná funkcionálna	23
3.4.2	Mikrokrokovanie	24
3.4.3	Zánik prúdu.....	25
3.5	Komerčne dostupné integrované obvody.....	26
3.5.1	Texas Instruments DRV8833	26
3.5.2	STMicroelectronics L6208	27
3.5.3	Trinamic TMC2130.....	28
3.6	Doplňkové senzory na riadenie pohonov	29
3.6.1	Enkodéry.....	29
3.6.2	Koncové senzory	29
4	Návrh pohonnej jednotky	30
4.1	Koncepčné riešenie polohovacej platformy	30
4.2	Požadovaný krútiaci moment.....	31

4.3	Výber komponentov	33
4.3.1	Krokový motor	33
4.3.2	Radič motoru TMC2130.....	34
4.3.3	Enkodér a koncové senzory	34
5	Riadiaca jednotka	35
5.1	Vývojová jednotka	35
5.2	Prototypové riešenie riadiacej jednotky	36
6	Návrh komunikačného protokolu.....	37
6.1	Požiadavky na komunikačný protokol.....	37
6.2	Príkazová správa	38
6.3	Subscriber	39
6.4	Echo správa	39
7	Návrh riadiaceho algoritmu (firmware)	40
7.1	Požiadavky na riadiaci algoritmus	40
7.2	Knižnice na obsluhu komponentov.....	41
7.2.1	Spôsob vytvárania dátových štruktúr	42
7.2.2	Príklad konkrétnej dátovej štruktúry	43
7.2.3	Zoznam vytvorených knižníc	44
7.3	Štruktúra riadiaceho algoritmu	45
7.4	Deklarácia premenných a inicializácia firmwaru.....	46
7.5	Hlavný stavový automat	47
7.5.1	Začiatok cyklu stavového automatu	48
7.5.2	Blokovanie zariadenia	49
7.5.3	Užívateľské riadenie	49
7.5.4	Nastavovanie parametrov	49
7.5.5	Koniec cyklu.....	49
7.6	Komunikácia pomocou BLE modulu	50
7.6.1	Prijímanie príkazových správ	50
7.6.2	Tvorba odosielaných echo správ	50
7.7	Riadenie krokových motorov.....	51
7.7.1	Komunikácia s radičom TMC2130	52
7.7.2	Odmeriavanie polohy	55
7.7.3	Stavový automat krokového motora.....	57
7.8	Komunikácia s pamäťou FRAM.....	59
7.8.1	Obsluha pamäte FRAM	59

7.8.2	Princíp adresovania.....	59
7.9	Integrácia doplnkových komponentov.....	60
7.9.1	Riadenie GPIO expandéru	60
7.9.2	IMU senzor	61
7.9.3	Výkonový modul	61
7.9.4	Odmeriavanie teploty	62
7.9.5	Užívateľské tlačidlo s RGB LED	62
8	Návrh užívateľského PC softwaru.....	63
8.1	Požiadavky na software	63
8.2	Rozloženie užívateľského prostredia	64
8.2.1	Dynamický riadiaci panel	65
8.2.2	Statický riadiaci panel	65
8.3	Vývojárske rozšírenie	66
8.4	Triedy užívateľského softwaru	67
8.5	Bluetooth komunikácia	68
8.6	Proces odoslania príkazovej správy	69
8.7	Proces vyhodnocovania echo správy	69
9	Testovanie navrhnutého riadiaceho algoritmu	70
9.1	Testovanie riadenia krokových motorov.....	71
9.1.1	Testovanie akcelerácie pri automatickom polohovaní	71
9.1.2	Testovanie akcelerácie pri manuálnom polohovaní	72
9.2	Testovanie zápisu do pamäte FRAM.....	73
9.3	Testovanie zápisu do registrov radiča TMC2130	73
9.4	Verifikácia softwarových funkcií.....	74
9.4.1	Simulácia prijímania správy	74
9.4.2	Vykresľovanie zvolených parametrov	74
10	Záver.....	75
	Bibliografia.....	77
	Zoznam obrázkov	81
	Zoznam tabuliek	83
	Zoznam skratiek	83
	Zoznam príloh.....	83

1 Úvod

Polohovanie objektov je neoddeliteľnou súčasťou rady procesov v rôznych odvetviach priemyslu. Rastúce požiadavky na efektivitu a bezpečnosť práce prinášajú stále častejšiu potrebu nahrádzať manuálne polohovanie poloautomatickými, prípadne plne automatickými manipulátormi. V závislosti na druhu ich aplikácie je potrebné zohľadniť viacero parametrov, ako sú napríklad veľkosť a tvar pracovného priestoru, požadovaná nosnosť, dynamické požiadavky alebo presnosť polohovania.

Moderné manipulátory disponujú viacerými stupňami voľnosti. Vďaka tomu umožňujú transláciu a rotáciu objektov vo viacerých smeroch priestoru. V aplikáciách, kde je dostatočne objekt polohovať v horizontálnej rovine, môžu byť robustné manipulátory nahradené kompaktnými polohovacími platformami.

Čoraz častejšie sa stretáme s aplikáciou polohovacích platforiem v rôznych vedecko-výskumných odvetviach. V prípade, že sú platformy využívané na cyklicky opakujúcu sa činnosť, je na mieste od nich vyžadovať dobrú opakovateľnosť. Taktiež sa môžeme stretnúť s aplikáciami, ktoré vyžadujú vysokú presnosť polohovania a často disponujú malým manipulačným priestorom. Príkladom je využitie platformy v mikroskopických systémoch.

Špecifickou oblasťou aplikácie polohovacích platforiem je možné označiť zariadenia počítačovej tomografie. Tieto zariadenia pracujú na princípoch röntgenového žiarenia a slúžia na vyšetrenie vnútornej štruktúry sledovaných objektov. Počas akvizície dát zariadenie postupne rotuje so sledovaným objektom okolo vertikálnej osi. S cieľom dosiahnutia dobrého výsledku je preto potrebné objekt v rámci zariadenia pred spustením akvizície polohovať v horizontálnej rovine tak, aby bola jeho vertikálna osa zosúladená s osou rotácie zariadenia. Postupná implementácia počítačovej tomografie do priemyselnej výroby prináša potrebu zvyšovať efektivitu práce operátorov týchto zariadení. Práve z toho dôvodu je vhodné na manipuláciu so sledovanými objektami využívať polohovacie platformy.

Na trhu existuje široká škála polohovacích platforiem. Pri aplikácii v špecifických oblastiach, kde dostupné riešenia narážajú na niektoré zo svojich limitov, preto nie je nutné zariadenie konštruovať od základu. S cieľom dosiahnutia špecifických požiadaviek je dostatočne navrhnuť riadiaci systém, ktorý je možné implementovať do existujúcich konštrukčných riešení.

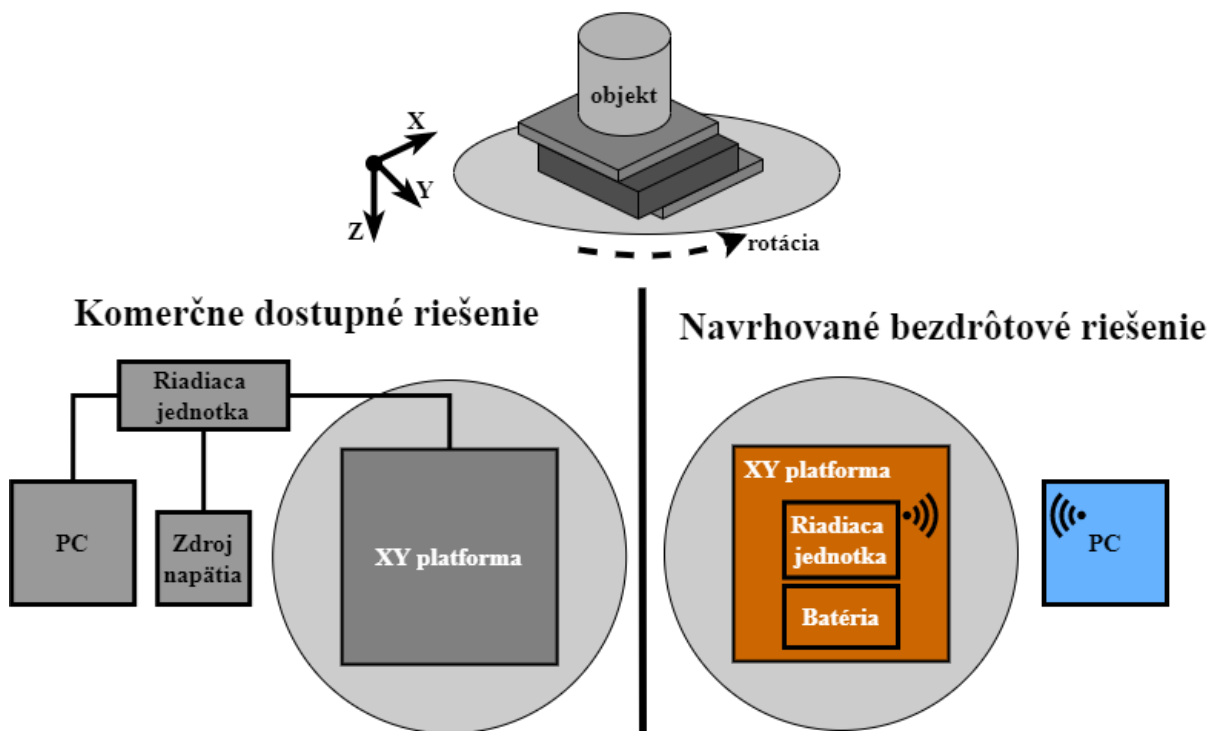
2 Definícia problémovej situácie a stanovenie cieľov práce

Polohovacie platformy využívajú rôzne druhy pohonov, ako napríklad lineárny motor, krokový motor, alebo servomotor. V aplikáciách vyžadujúcich vysokú presnosť polohovania môžu byť využité benefity piezomotorov, schopných polohovať objekty v rádoch nanometrov.

Za účelom riadenia motorov je v každom prípade potrebné ich pripojenie ku zdroju napätia a vhodnej riadiacej jednotke. Komerčne dostupné platformy na tento účel využívajú externé komponenty, s ktorými sú prepojené pomocou káblových zväzkov.

Externé komponenty spolu s prípojnou kabeľazou prinášajú problémy pri aplikáciách v obmedzených priestoroch, ako sú napríklad mikroskopy. Taktiež je problematické využitie v zariadeniach počítačovej tomografie, kde prichádza ku rotácii platformy, ako je popísané v úvode tejto práce.

Za účelom eliminácie týchto problémov, je potrebné navrhnuť riadiaci systém, ktorého schému uvádza Obr. 2-1. Tento systém je napájaný batériou, disponuje bezdrôtovou komunikáciou a kompaktnou riadiacou jednotkou, pre ktorú je nutné navrhnuť riadiaci firmware a užívateľský software. Jednou z hlavných požiadaviek kladených na navrhovaný riadiaci systém je jeho univerzálnosť, ktorá prináša možnosť prispôsobenia na rôzne aplikácie. Z toho dôvodu je pri návrhu potrebné dbať na minimalizáciu riadiacej jednotky, jednoduché prispôsobenie výkonovej elektroniky, parametrov riadiaceho firmwaru a softwaru. Vhodné je navrhnuť riešenie firmwaru tak, aby mohlo byť jednoducho rozšíriteľné o ďalšie funkcionality.



Obr. 2-1 Navrhované bezdrôtové riešenie

2.1 Členovia tímu a rozdelenie úloh

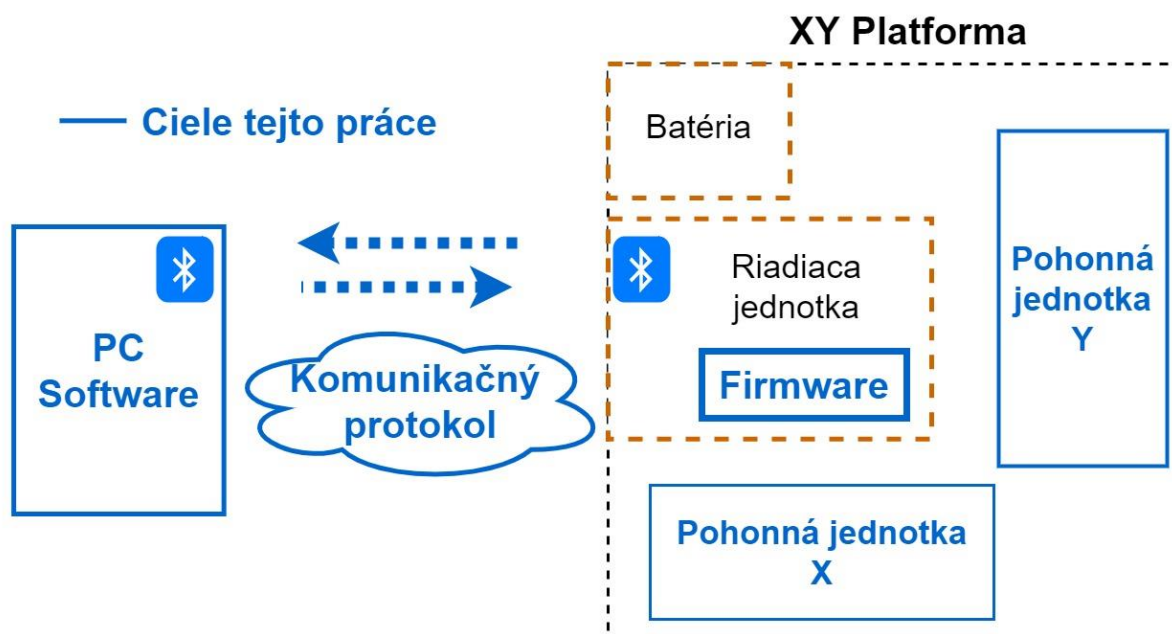
S ohľadom na náročnosť úlohy bolo riešenie pridelené dvojici študentov. Keďže sa jeho jednotlivé časti prekrývajú, alebo vzájomne na seba nadväzujú, je potrebné počas návrhu riešenia dbať na dodržiavanie stanoveného vývojového plánu a intenzívnu vzájomnú spoluprácu s cieľom dosiahnutia dobrého výsledku.

Bc. Matúš Remiš (autor)

Cieľom tejto práce je analýza dostupných polohovacích platforiem a možností ich riadenia, ktoré sú popísané v kapitole 3. Získané poznatky sú následne ďalej v práci využité v kapitole 4, pri voľbe komponentov testovacej pohonnej jednotky. Na zabezpečenie pravidiel komunikácie medzi riadiacou jednotkou uvedenou v kapitole 5 a užívateľským PC, je potrebné definovať komunikačný protokol, ktorého návrhom sa zaoberá kapitola 6. Ako vyplýva z názvu tejto diplomovej práce, hlavným cieľom je návrh firmwaru pre riadiacu jednotku. Popisu jeho štruktúry a jednotlivých funkcionalít sa venuje kapitola 7. Keďže je potrebné zabezpečiť možnosť riadenia systému z PC, kapitola 8 popisuje návrh a vizualizáciu užívateľského softwaru. Výsledky testovania navrhnutého firmwaru a softwaru sú uvedené v kapitole 9. Schému cieľov tejto práce je možné vidieť na Obr. 2-2.

Bc. Adam Ondryáš [39]

Ondryáš sa vo svojej práci zaoberá návrhom a výrobou riadiacej jednotky disponujúcej bezdrôtovou komunikáciou. Vo svojej práci taktiež rieši konceptualizáciu funkcií riadiacej jednotky, voľbu doplnkových senzorov, návrh výkonovej elektroniky, voľbu batérie a návrh jej nabíjacieho obvodu.



Obr. 2-2 Schéma cieľov práce

3 Polohovacie platformy a ich riadenie

Účelom tejto časti práce je získanie prehľadu v oblasti polohovacích platforiem a ich riadenia. Podkapitoly 3.1 sa zaoberajú popisom základných typov lineárnych pojazdvých mechanizmov.

Prehľad komerčne dostupných riešení polohovacích platforiem, ktorý slúži ako inšpirácia pri návrhu univerzálneho systému je uvedený v podkapitole 3.2. Keďže krokové motory patria medzi často aplikované pohony v polohovacích platformách, podkapitola 3.3 sa zaoberá popisom princípu ich funkcie a rozdelením na základe ich konštrukcie a vinutia.

Hlavným cieľom tejto práce je návrh riadiaceho algoritmu polohovacej platformy. Podkapitola 3.4 preto popisuje princípy riadenia krokových motorov a taktiež funkcionality ich moderných radičov. Príklady a porovnanie komerčne dostupných radičov krokových motorov sú uvedené v podkapitole 3.5.

Táto kapitola je ukončená prehľadom doplnkových komponentov pohonnej jednotky, ktorými sú enkodér a koncové senzory. Popisom ich jednotlivých typov sa zaoberá podkapitola 3.6.

3.1 Pojazdové mechanizmy osí

Pojazdové mechanizmy slúžia na sprostredkovanie translačného pohybu v jednej ose. Na základe princípov ich funkcie a využívaných komponentov je možné pojazdvé mechanizmy rozdeliť do kategórií:

- Mechanizmus s lineárnym motor (3.1.1)
- Mechanizmus využívajúci piezomotor (3.1.2)
- Mechanizmus kombinujúci rotačný motor a ozubený hrebeň (3.1.3)
- Mechanizmus kombinujúci rotačný motor a ozubený remeň (3.1.4)
- Mechanizmus kombinujúci rotačný motor a pohybovú skrutku s maticou (3.1.5)

3.1.1 Lineárny motor

Princíp funkcie lineárneho motoru uvedeného na Obr. 3-1, je možné odvodiť z princípu rotačných motorov, keďže v oboch prípadoch prichádza ku elektromagnetickej interakcii medzi cievkami vinutia a permanentnými magnetmi. Na rozdiel od rotačných motorov je u lineárnych motorov často uvádzaná do pohybu napájaná časť, zatiaľ čo elektricky pasívna časť ostáva bez pohybu. Pojmom „elektricky pasívna“ je označená časť motoru, ktorá obsahuje permanentné magnety. Výhodou tohto typu je jeho schopnosť dosahovať veľké zrýchlenia a taktiež vysokú rýchlosť pohybu, vysoké rozlíšenie a nízke potreby na údržbu. Medzi výraznú nevýhodu patrí však jeho vysoká cena. [1],[2]

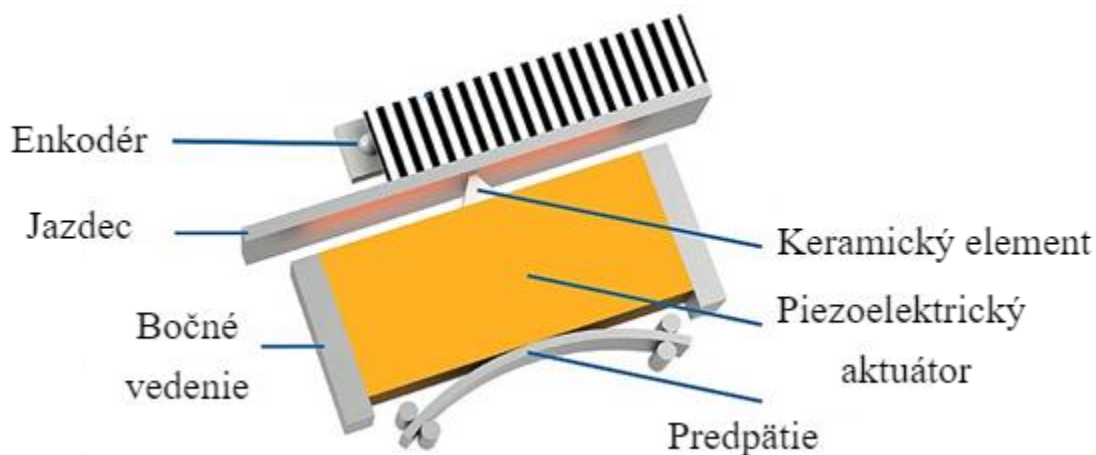


Obr. 3-1 Lineárny motor [1]

3.1.2 Piezomotor

Na pohyb piezomotorov je využívaný piezoelektrický jav v kombinácii s trením. Tieto pohony sa vyznačujú predovšetkým veľmi malým jednotkovým krokom, samosvornosťou, vysokou hustotou výkonu, adaptivitou a kompaktnosťou. Nevýhodou je však pomerne náročné riadenie, potreba senzorov s vysokým rozlíšením a krátky rozsah pohybu. Z toho dôvodu sa piezomotory používajú prevažne v špecifických polohovacích aplikáciách. [3]

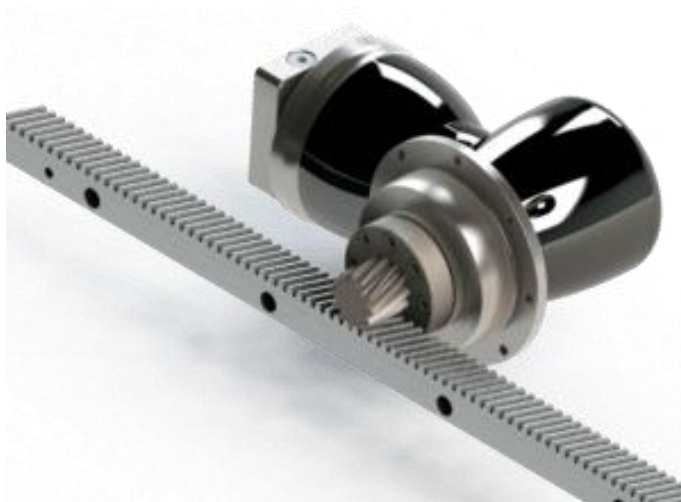
Medzi známymi výrobcami polohovacích systémov integrujúcimi piezomotory patrí americký výrobca PI-Physik Instrumente. Princíp pohonu uvedeného na Obr. 3-2 je založený na oscilácii piezoelektrického aktuátora. Úlohou keramického elementu je prenos týchto oscilácií na jazdec, ktorý je hlavným pohyblivým elementom tohto mechanizmu.



Obr. 3-2 Časti piezomotoru - upravené z [4]

3.1.3 Kombinácia rotačného motora a ozubeného hrebeňa

Tento mechanizmus zabezpečuje translačný pohyb prevodom rotačného pohybu motora pomocou ozubeného hrebeňa a pastorku. Pastorok je často s motorom prepojený pomocou prevodovky, ktorá znižuje rýchlosť a zvyšuje dosiahnuteľný krútiaci moment. Medzi hlavné výhody tohto mechanizmu patrí možnosť jeho prakticky neobmedzeného predlžovania. Ako nevýhodu je možné považovať jeho vyššiu hlučnosť v porovnaní s ostatnými mechanizmami.[5]



Obr. 3-3 Mechanizmus kombinujúci pastorok a ozubený hrebeň [6]

3.1.4 Kombinácia rotačného motora a ozubeného remeňa

Lineárny pohyb u pojazdového mechanizmu na Obr. 3-4 je zabezpečený rozpohybovaním ozubeného remeňa pomocou rotačného motora. Medzi výhody tohto riešenia patrí jeho tichý chod. Typické dosahované rýchlosti sú 3 až 5 m/s. Keďže počas prevádzky prichádza ku postupnej plastickej deformácii remeňa, je potrebné pravidelne kontrolovať jeho predopnutie. Nevýhodou je taktiež požiadavka vyššieho krútiaceho momentu pohonov v porovnaní s mechanizmami využívajúcimi pohybové skrutky. [7][8]



Obr. 3-4 Mechanizmus využívajúci ozubený remeň [1]

3.1.5 Kombinácia rotačného motora a pohybovej skrutky

Princíp tohto mechanizmu je založený na prevode rotačného pohybu motora na translačný pohyb jazdca pomocou pohybovej skrutky zo závitom. Podľa typu aplikácie je možné využiť guľôčkovú pohybovú skrutku, alebo skrutku s trapézovým závitom.

V praxi nájdú najčastejšie uplatnenie guľôčkové skrutky. Medzi ich hlavné výhody patrí široká škála povoleného zaťaženia, hladký chod, výborná presnosť a opakovateľnosť. Taktiež sa vyznačujú pomerne jednoduchou implementáciou do systému a dobrou životnosťou komponentov. Medzi nevýhody patrí relatívne vysoká cena, možný výskyt rezonancie v prípade veľkého zdvihu alebo vysokých otáčok. V prípade potreby vysokého zdvihu je možné nahrádzať guľôčkové pohybové skrutky mechanizmami kombinujúce pastorok a ozubený hrebeň.[9][10]

Pri cenovo kritických aplikáciách, ktoré nevyžadujú vysokú presnosť je možné nahrádzať guľôčkové skrutky pohybovými skrutkami s trapézovým závitom. Medzi ich hlavné výhody patrí dobrá cenová dostupnosť a taktiež možnosť využitia samosvornosti, čo eliminuje potrebu návrhu brzdy vypnutého pohonu. Pri voľbe stúpania závitú skrutky je potrebné zohľadniť jeho vplyv na efektívnosť mechanizmu. Keďže prichádza ku priamemu kontaktu skrutky a matice, je taktiež potrebné zvoliť dobrú kombináciu ich materiálov, ktoré kriticky ovplyvňujú vzájomné trenie. Pri horizontálnych aplikáciách požadujúcich dobrú presnosť a opakovateľnosť je vhodné využiť matice s možnosťou predopnutia, ktoré zabezpečia lepšie vymedzenie vôlí, čo vedie ku zvýšeniu presnosti polohovania.[9][11]

Prepojenie medzi motorom a pohybovou skrutkou môže byť zabezpečené pomocou prevodovky, alebo priameho prepojenia spojkou. Pri využití prevodovky je možné zvýšiť stúpanie závitú skrutky a požadovaný prevodový pomer dosiahnuť správnou voľbou parametrov prevodovky. Vďaka tomu je možné znížiť rotačnú rýchlosť skrutky pri udržaní požadovanej lineárnej rýchlosti posunu, čoho výsledkom je zníženie vplyvu zotrvačnosti skrutky a taktiež zníženie hluku. V prípade, že aplikácia nevyžaduje integráciu prevodovky, je možné využiť priame spojenie motoru a pohybovej skrutky pomocou spojky. Na trhu existujú ich rôzne typy. Keďže pevné spojky vyžadujú vysokú presnosť montáže, často je možné sa stretnúť s využitím rôznych typov flexibilných spojok. Na Obr. 3-5 je uvedený motor, ktorý nahrádza spojku priamou integráciou pohybovej skrutky.[9]



Obr. 3-5 Rotačný motor s integrovanou pohybovou skrutkou [12]

3.2 Analýza komerčne dostupných polohovacích platforiem

Za účelom polohovania vo dvoch osiach horizontálnej roviny je možné využiť ortogonálnu kombináciu dvoch lineárnych systémov. Z dôvodu nutnosti využitia prepojovacích adaptérov sa často stretáme s integráciou dvoch lineárnych systémov do kompaktných konštrukčných riešení, ktorým sú polohovacie platformy. Nasledujúce podkapitoly sa preto zaoberajú krátkou analýzou ich komerčne dostupných riešení.

3.2.1 Výrobcovia polohovacích platforiem

Vývojom motorizovaných polohovacích platforiem sa vo svete zaoberá pomerne veľké množstvo firiem. Zákazník si podľa svojich priorít môže vybrať zo širokého spektra produktov od jednoduchých relatívne lacných, až po sofistikované platformy. Tie z pravidla disponujú predovšetkým veľkou presnosťou, tuhosťou, nosnosťou a rôznymi inými vlastnosťami, vďaka ktorým môžu byť použité aj na veľmi precízne a špecifické úlohy.

Medzi známe spoločnosti zaoberajúce sa vývojom a distribúciou polohovacích platforiem patria napríklad :

- CEDRAT TECHNOLOGIES
- SmarAct GmbH
- IntelLiDrives, Inc.
- Physik Instrumente (PI) GmbH & Co. KG
- Newport Corporation
- Thorlabs
- Sigmakoki
- Standa
- Optimal Engineering Systems Inc.

3.2.2 Typy využívaných pohonov

Voľba pohonu výrazne vplýva na vlastnosti, výkon no predovšetkým cenu polohovacej platformy. Tab. 3-1 uvádza porovnanie platforiem výrobcu Newport v závislosti na type zvoleného pohonu.

Tab. 3-1 Porovnanie pohonov polohovacích platforiem [13]

Typ motoru	Rozsah pohybu	Rýchlosť	Rozlíšenie	Opakovateľnosť	Zátťaž	Cena
Lineárny motor	Dlhý	Vysoká	Vysoké	Vysoká	Vysoká	Vysoká
BLDC motor	Stredný	Stredná	Stredné	Vysoká	Vysoká	Stredná
DC motor	Stredný	Stredná	Stredné	Stredná	Vysoká	Stredná
Krokový motor	Stredný	Stredná	Stredné	Stredná	Stredná	Nízka
Piezo motor	Krátky	Nízka	Vysoké	Nízka	Vysoká	Stredná

3.2.3 Komerčne dostupné polohovacie platformy

Manipulátor od spoločnosti *IntelLi Drivers, Inc* na Obr. 3-6 je vybavený guľôčkovou skrutkou so stúpaním 4mm, na ktorej pohon si zákazník môže vybrať medzi krokovým motorom Nema 23 a servomotorom s enkodérom. Rozsah pohybu je 200 x 200 mm s presnosťou opakovania $\pm 5 \mu\text{m}$. Výrobca uvádza nosnosť platformy 30 kg a maximálnu rýchlosť pohybu 40 mm/s v prípade využitia krokového motoru, resp. 80mm/s v prípade využitia servo pohonu. [14]



Obr. 3-6 Platforma XY-BSMA [14]

Platformu od spoločnosti *Physik Instrumente (PI) GmbH & Co. KG* na Obr. 3-7 je možné zakúpiť v prevedení s dvojfázovým krokovým motorom, vďaka ktorému je možné dosiahnuť vysoký krútiaci moment aj pri nižších otáčkach a taktiež vyššie rozlíšenie. Druhý variant je vybavený DC motorom, ktorý zabezpečuje dosiahnutie vyšších rýchlostí a zníženie vibrácií. Taktiež je možné zvoliť lineárny enkodér s rozlíšením 1nm, alebo rotačný enkodér s rozlíšením 20000 pulzov na otáčku. Pri využití krokového motoru zariadenie dosahuje maximálnu rýchlosť 45 mm/s a presnosť $\pm 0.5 \mu\text{m}$. [15]



Obr. 3-7 L-731 Precision XY Stage [15]

Väčšina polohovacích systémov od spoločnosti *Newport Corporation* zabezpečuje lineárny pohyb iba v jednom smere. Aby sa predišlo nutnosti kombinácie dvoch lineárnych systémov pomocou rôznych adaptérov, uviedla spoločnosť na trh radu produktov ONE-XY. Táto rada integruje možnosť pohybu v oboch osiach do jedného kompaktného zariadenia za účelom zvýšenia presnosti a tuhosti systému, vďaka čomu nájdú tieto platformy uplatnenie v rade dynamických aplikácií. Produkt ONE-XY200 disponuje rozsahom pohybu 190 mm v oboch osiach s presnosťou $\pm 2.5\mu\text{m}$. Maximálna rýchlosť 200mm/s je zabezpečená lineárnym motorom. Výrobca taktiež okrem iného uvádza nosnosť zariadenia 15 kg. [16]



Obr. 3-8 Platforma ONE-XY200 [16]

Platforma od spoločnosti Standa na Obr. 3-9 disponuje rozsahom pohybu 200mm v oboch osiach. Gul'ôčková skrutka so stúpaním 2mm zabezpečuje presnosť polohovania $\pm 2.5\mu\text{m}$. Podľa konkrétneho typu je pohon zabezpečený bipolárnym krokovým motorom, alebo trojfázovým BLDC motorom. Výrobca uvádza maximálnu rýchlosť 30-50mm/s pri zaťažení 4kg v závislosti od konkrétneho typu zariadenia. Maximálna nosnosť je 8kg. [17]



Obr. 3-9 Platforma 8MT200XY [17]

3.2.4 Vhodný typ pohonu testovacej jednotky

Medzi požiadavky kladené na navrhovaný riadiaci systém patrí univerzálnosť, možnosť jednoduchého riadenia a nízka cena. Na základe vplyvu typov pohonov na vlastnosti platformy uvedených v Tab. 3-1 a analýze komerčne dostupných platforiem, je možné považovať kombináciu krokového motora a pohybovej skrutky za ideálny kompromis pri návrhu univerzálnej cenovo dostupnej testovacej pohonnej jednotky.

3.3 Krokové motory

Krokové motory (KM) patria medzi široko dostupné pohony bežne využívané v polohovacích aplikáciách. Rotor KM sa pohybuje v diskretných úsekoch – krokoch. Zatiaľ čo u najjednoduchších motorov je veľkosť kroku 90° , štandardne využívané motory disponujú 200 krokmi na otáčku, čo odpovedá veľkosti kroku 1.8° . Postupným napájaním cievok motora v jednotlivých krokoch je možné jednoducho riadiť smer otáčania, rýchlosť otáčok a presnú polohu hriadeľa motora. Reguláciu rýchlosti a zrýchlenie KM je možné dosiahnuť riadením časového úseku medzi jednotlivými krokmi. V porovnaní s klasickým DC motorom, na ktorého riadenie je potrebná spätná väzba pomocou enkodéru, poskytujú KM výhodu v podobe možnosti riadenia bez spätnej väzby.[18]

V porovnaní s väčšinou motorov, ktoré majú pri nízkych rýchlostiach malý krútiaci moment, majú KM výhodu v podobe pôsobiaceho magnetického poľa v každom kroku, vďaka čomu dosahujú vysoký moment aj v nízkych rýchlostiach a taktiež počas zastavenia. Vďaka tejto vlastnosti sa jedná o vhodný pohon na polohovanie pri nízkych rýchlostiach. Medzi ďalšie výhody KM patrí jednoduché a cenovo dostupné riadenie digitálnymi pulzami a taktiež schopnosť udržiavať polohu bez potreby riadenia. Jednoduchá konštrukcia priaznivo vplýva na jeho životnosť.[18]

Medzi nevýhody krokových motorov patria vibrácie, ktoré významne zvyšujú hlučnosť motora, strata kroku, ktorá môže nastať pri preťažení motora, strata krútiaceho momentu pri vyšších otáčkach a problémy spojené s rezonanciou. Medzi čiastočné riešenie týchto nevýhod patrí využitie moderných radičov, schopných redukovať niektoré zo spomenutých javov.[18]

3.3.1 Typy krokových motorov

Vlastnosti KM ako rozlíšenie (veľkosť kroku), maximálna rýchlosť a krútiaci moment, sú ovplyvňované ich vnútornou konštrukciou, ktorá taktiež určuje spôsob riadenia. Podľa konfigurácie rotora a statora je možné rozdeliť KM do troch skupín [19]:

- KM s variabilnou reluktanciou
- KM s permanentným magnetom
- Hybridný KM

Vďaka tomu, že hybridné KM kombinujú vlastnosti KM s variabilnou reluktanciou a permanentným magnetom, sú často využívané v moderných polohovacích aplikáciách. Bližší popis hybridných KM spolu s možnosťami ich riadenia je uvedený v nasledujúcich podkapitolách.

3.3.2 Hybridný krokový motor

Hybridný KM kombinuje princípy KM s variabilnou reluktanciou a permanentným magnetom, za účelom dosiahnutia malého uhlu kroku a vysokého momentu pri malých rozmeroch.

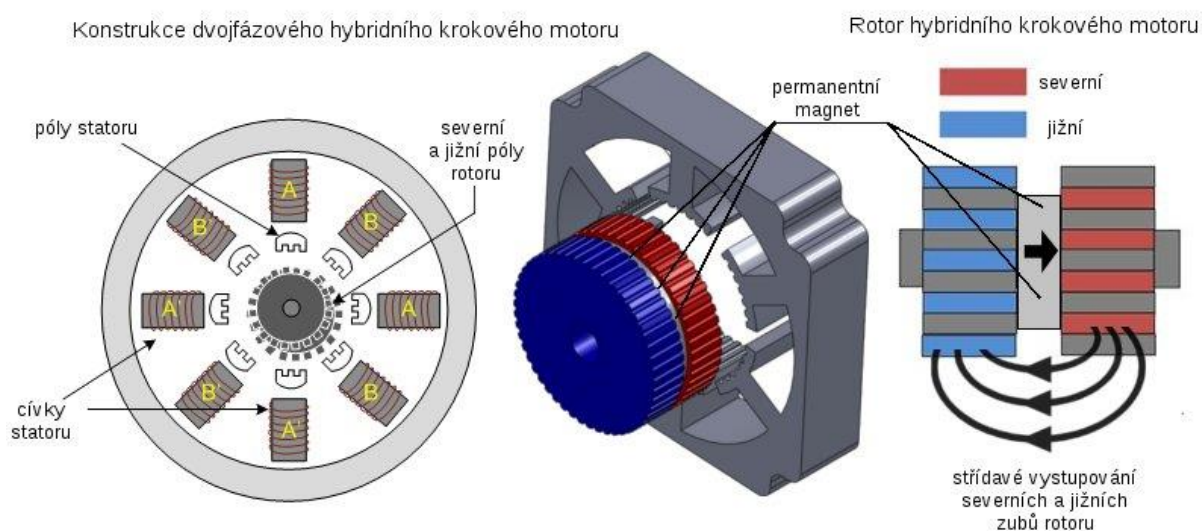
Jadro statoru je rovnaké ako u KM s variabilnou reluktanciou. Rotor sa však skladá z valcového permanentného magnetu, ktorého oba póly sú zakončené ozubeným pólovým nástavcom z magneticky mäkkého materiálu, ako je uvedené na Obr. 3-10. Zuby sú na opačných pólových nástavcoch voči sebe natočené o polovicu zubového rozstupu. Ďalší princíp funkcie je podobný ako u krokového motoru s permanentným magnetom, kedy postupným napájaním jednotlivých fáz vinutia statoru, prichádza ku zmene magnetickej polarity jeho pólů. Snahou rotora je zosúladiť svoju orientáciu s orientáciou magnetického toku statorového vinutia, čo vedie ku zmene polohy rotora. Bežne využívané hybridné KM disponujú veľkosťou kroku 1.8° , prípadne 0.9° . [20]

Výhody hybridného KM [23]:

- Možnosť dosahovať malé kroky
- Vyšší moment v porovnaní s KM s variabilnou reluktanciou
- Zbytkový moment – schopnosť udržať polohu
- Vyššia efektívnosť pri nízkych rýchlostiach

Nevýhody hybridného KM [23]:

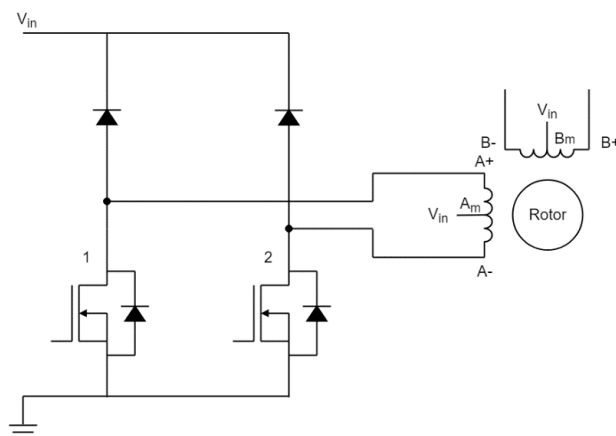
- Väčšia hmotnosť z dôvodu magnetov na rotore
- Vyššia cena v porovnaní s KM s variabilnou reluktanciou
- Vyššia zotrvačnosť



Obr. 3-10 Schéma hybridné krokového motoru [22]

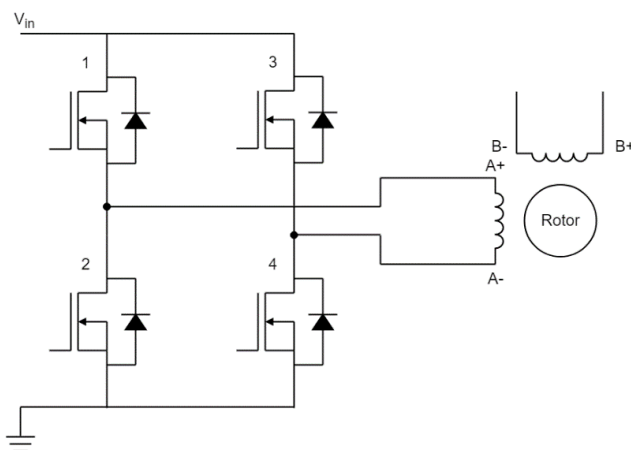
3.3.3 Vinutia krokových motorov

Unipolárne KM disponujú 2 cievkami, ktoré sú identické a nie sú elektricky spojené. Ako uvádza schéma na Obr. 3-11, každá z cievok má vývod vychádzajúci z jej stredu. Pri unipolárnom riadení je na vinutie pripájané vždy len napätie jednej polarita. Výhodou riadenia je potreba polovičného počtu tranzistorov v porovnaní s riadením bipolárneho motora. Stredné vodiče sú vždy pripojené ku kladnému pólu zdroja. Následne je možné uzemňovaním jednotlivých koncov cievok dosiahnuť požadovanú sekvenciu na otáčanie motora.[20]



Obr. 3-11 Unipolárny krokový motor [20]

Bipolárne motory sa z hľadiska vyhotovenia veľmi podobajú na unipolárne. Podstatným rozdielom je však absencia stredového vývodu. Vplyvom tejto odlišnosti vyžadujú bipolárne motory iný typ riadenia ako unipolárne. Potrebne je zabezpečenie zmeny toku prúdu cievkami, ktorá je dosiahnutá zmenou polarita. Z toho dôvodu je potrebné ich zapojenie v tzv. H-mostíkoch zobrazených na Obr. 3-12. Vďaka využitiu celej cievky, nie len jej polovice ako u unipolárnych motorov, disponujú bipolárne motory schopnosťou dosiahnuť väčší moment.[20]



Obr. 3-12 Bipolárny krokový motor[20]

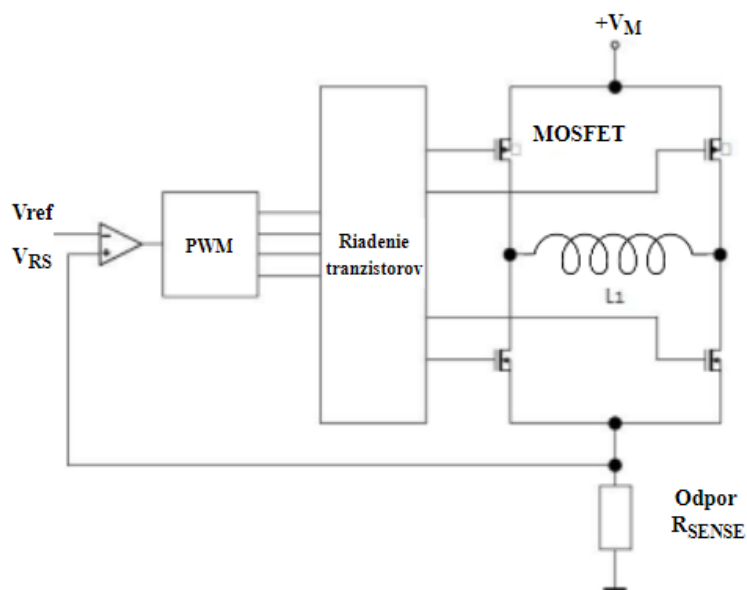
3.4 Radiče krokových motorov

Krokové motory na svoj chod vyžadujú externé komponenty. Z pravidla sa jedná o zdroj napätia, logický sekvenčný obvod, spínacie prvky a zdroj riadiacich pulzov definujúcich frekvenciu krokov. V dnešnej dobe sú dostupné riadiace čipy, ktoré združujú riadiace komponenty do kompletného integrovaného obvodu.[23]

3.4.1 Základná funkcionálna

Moderné radiče krokových motorov fungujú na princípe dodávania konštantného budiaceho prúdu do cievok motora. Obr. 3-13 zobrazuje základnú schému riadenia prúdu v cievke motora. Prúd je u týchto radičov regulovaný vďaka meraniu napät'ového úbytku na odpore zapojeného v sérii s cievkou krokového motora. Pri náraste prúdu rastie napät'ový úbytok, ktorý je interným komparátorom porovnávaný s referenčnou hodnotou napätia. V prípade, že príde k jej dosiahnutiu, je napájacie napätie odpojené až do momentu, kedy je potrebné vykonať ďalší krok. Týmto spôsobom prúd rastie a klesá v závislosti na pripojení, resp. odpojení napätia, čo vedie ku udržaniu priemerného požadovaného prúdu počas jednotlivých krokov motora.[24]

Výhodou týchto radičov je možnosť dosahovať vyššiu strmosť prúdu, keďže sú z pravidla napájané vyšším napätím ako klasické L/R radiče. Taktiež je možné vďaka vyššiemu napätiu dosahovať vyššiu hodnotu prúdu, čo priaznivo vplýva na veľkosť dosiahnuteľného krútiaceho momentu.[24]

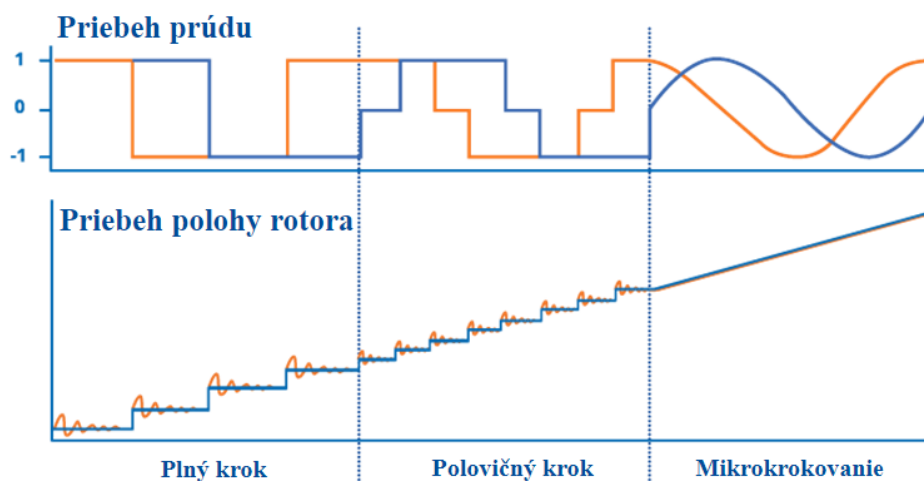


Obr. 3-13 Schéma odmeriavania prúdu v cievke motora, upravené z [27]

3.4.2 Mikrokrokovanie

Pri riadení krokových motorov často narážame na problémy spojené s ich diskretným pohybom po jednotlivých krokoch. Táto vlastnosť vedie k rade nepriaznivých následkov, ako znížená hladkosť pohybu, zvýšená hlučnosť z dôvodu vibrácií, výrazné prúdové špičky, alebo rezonancia. Za účelom eliminácie týchto vlastností je vhodné využívať radiče KM disponujúce možnosťou mikrokrokovania. [25]

Princíp je založený na postupnom budení cievok motora ako je uvedené na Obr. 3-14, kedy je obdĺžnikový priebeh prúdu nahradený sínusovým. Vďaka tejto zmene je rotor schopný dosahovať polohy medzi dvoma krokmi. U moderných radičov sa často stretávame s možnosťou využitia 1-256 mikrokrokov na jeden plný krok. Pri riadení bežného KM disponujúcim veľkosťou kroku $1,8^\circ$, je preto možné dosiahnuť rozlíšenie až 51200 mikrokrokov na otáčku.[26]



Obr. 3-14 Porovnanie vplyvu mikrokrokovania [27]

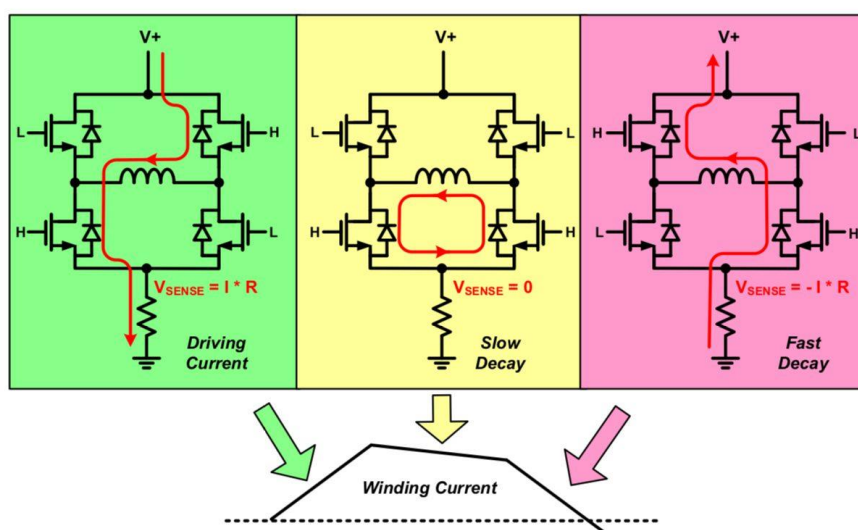
Tento spôsob riadenia prináša radu výhod. Jedná sa predovšetkým o plynulejší chod, výrazné zníženie prevádzkového hluku a zníženie problémov spojených s rezonanciou KM. Pri voľbe hodnoty mikrokrokov je však potrebné zohľadniť ich negatívny vplyv na krútiaci moment, kedy pri ich zvyšovaní prichádza k poklesu inkrementu krútiaceho momentu, čoho následkom môže byť za určitých podmienok vznik nepresností polohovania. Tejto problematike sa venujú [25] a [26]. Analýzou vplyvu mikrokrokov na presnosť polohovania u dostupných radičov sa zaoberá [28].

3.4.3 Zánik prúdu

Počas chodu KM prichádza ku kontinuálnemu striedaniu dvoch stavov napájania cievky. Počas doby “On time”, prichádza ku pripojeniu cievky pomocou H-mostíku ku zdroju budiaceho napätia a prúd narastá. Naopak v čase “Off time”, kedy je cievka odpojená od napájania, prichádza k poklesu prúdu. S cieľom zabrániť poškodeniu riadiacich tranzistorov, je možné využiť paralelne pripojenie diód, slúžiacich na disipáciu prúdu. Za účelom zachovania kompaktnosti integrovaného obvodu a jeho nižšej ceny, je možné funkciu diód nahradiť vhodným riadením tranzistorov. Obr. 3-15 uvádza dva možné módy riadenia pomalého a rýchleho zániku.[29]

V móde pomalého zániku (Slow decay) prichádza k zopnutiu oboch tranzistorov v hornej alebo spodnej časti H-mostíku, následkom čoho prichádza ku skratovaniu príslušného fázového vinutia. Nastáva pomalý pokles prúdu definovaný L/R časovou konštantnou motora.[29]

Rýchly zánik (Fast decay) je založený na princípe zmene polarite budiaceho napätia zopnutím opačných tranzistorov, čo má za následok zmenu toku prúdu. Fast decay umožňuje motoru rýchlu reakciu na zmeny riadiacich pulzov, ale prináša pomalé zastavenie motoru. Nevýhodou tohto módu je výraznejšie zvlnenie prúdu.[29]



Obr. 3-15 Porovnanie typov zániku prúdu [29]

U moderných radičov krokových motor sa môžeme stretnúť s módom mixed decay, ktorého cieľom je kombinácia oboch módov za účelom aproximácie sinusového priebehu prúdu v cievke. Ako príklad je možné uviesť patentovanú technológiu SpreadCycle PWM Chopper od spoločnosti Trinamic. Výhodou tejto funkcie je zabezpečenie hladšieho priebehu prúdu, a taktiež hladšieho priebehu krútiaceho momentu.[30]

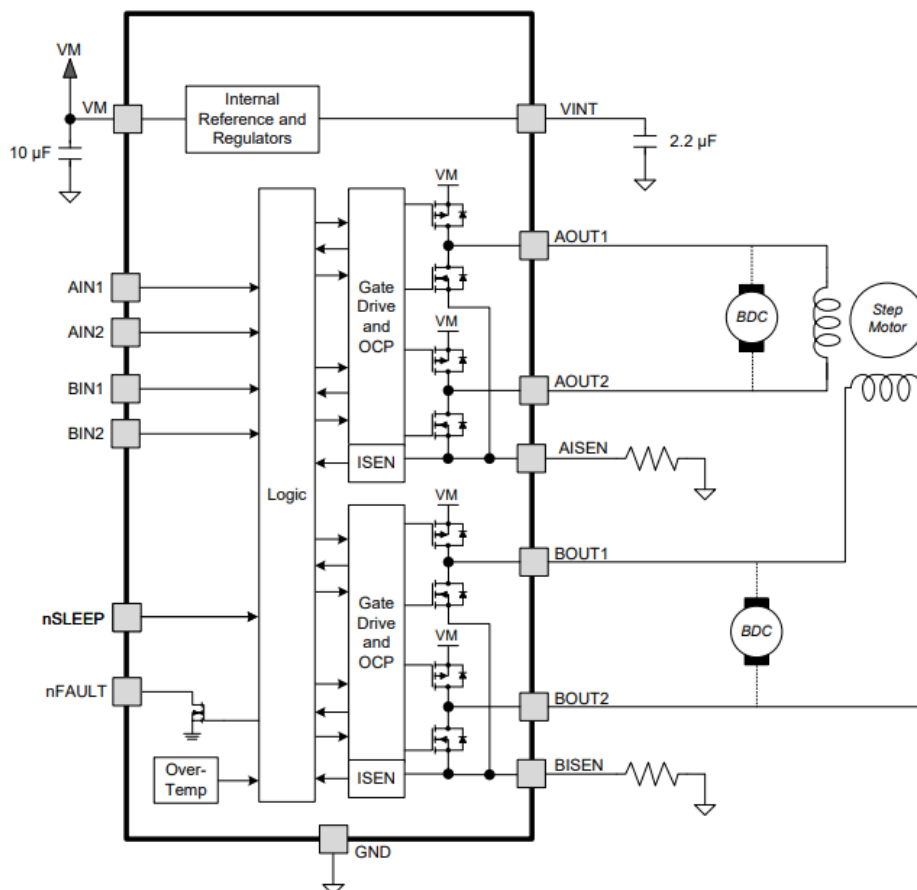
3.5 Komerčne dostupné integrované obvody

Na trhu existuje široká škála integrovaných obvodov slúžiacich na riadenie krokových motorov. Relatívne lacné obvody v sebe zahŕňajú iba základné funkcionality. Existuje však rada sofistikovaných obvodov poskytujúcich užívateľovi rozsiahle možnosti nastavovania riadiacich parametrov.

3.5.1 Texas Instruments DRV8833

Integrovaný obvod DRV8833 od spoločnosti Texas Instruments integruje dvojicu H-mostíkov a reguláciu prúdu motora. Je určený pre napájacie napätie 2,7 až 10,8 V a vie dodávať výstupný prúd až do 1,5 A. [31]

Riadenie je zabezpečené vstupnými PWM signálmi, vďaka čomu je možná jeho jednoduchá integrácia do riadiaceho systému. Regulácia prúdu je zabezpečená pomocou PWM modulácie s fixovanou frekvenciou pracujúcou v móde „slow decay“. Tento čip disponuje taktiež nízkonapäťovým spánkovým módom, vďaka čomu je možné dosiahnuť úsporu energie pri nečinnosti motora. [31]



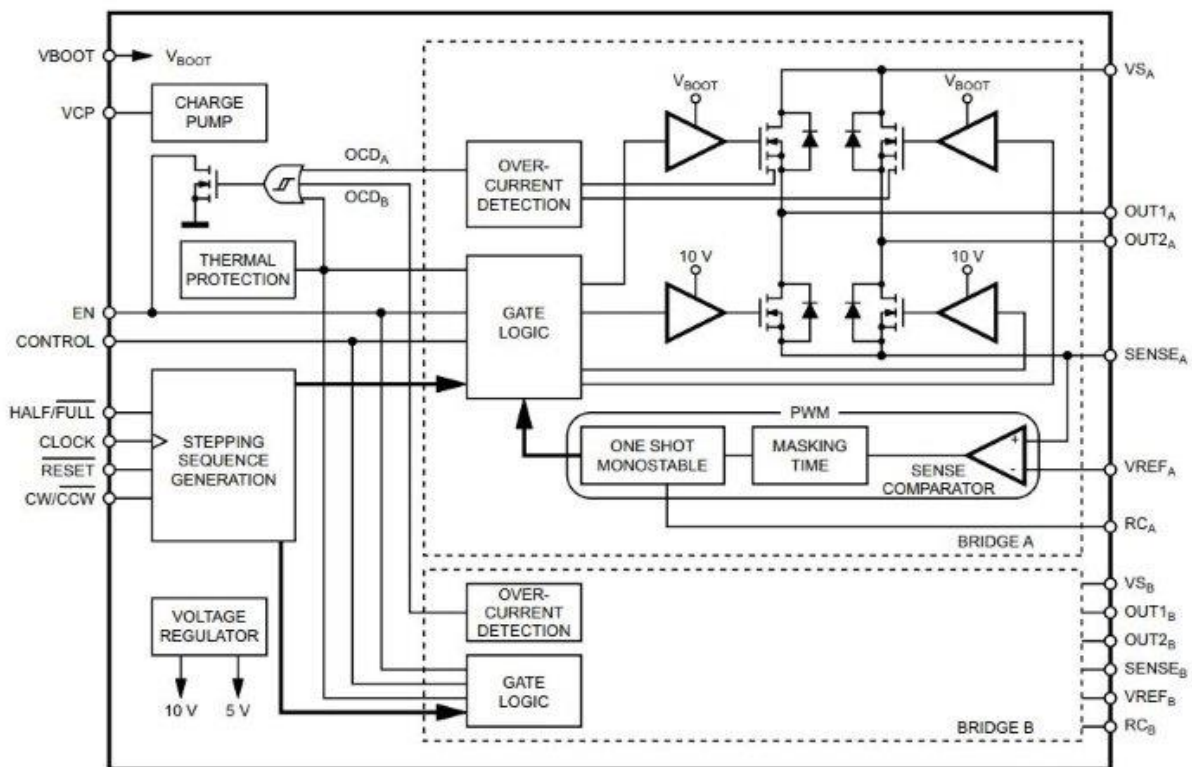
Obr. 3-16 Integrovaný obvod DRV8833 [31]

3.5.2 STMicroelectronics L6208

Výhodou čipu L6208, ktorého schéma je uvedená na Obr. 3-17 , je v porovnaní s DRV8833 predovšetkým integrovaný generátor spínacích sekvencií tranzistorov. Na rozdiel od predchádzajúceho riešenia, využíva preto L6208 ako zdroj riadiacich pulzov iba jeden digitálny vstup, vďaka čomu je možné jeho jednoduchšie riadenie. Pripojený KM vykoná v reakcii na nábežnú hranu Clock signálu podľa nastavenia jeden plný alebo polovičný krok. Taktiež je integrovaná dvojica nezávislých riadení prúdu vo fázových vinutiach motoru [32].

Vlastnosti:

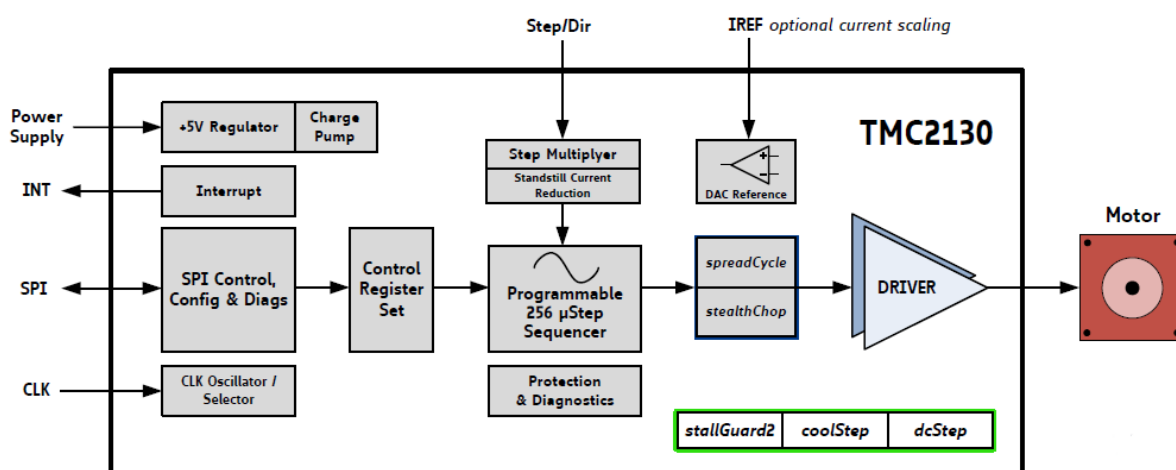
- Napájacie napätie 8-52 V
- 5.6 A amplitúda výstupného prúdu (2.8 A RMS)
- Maximálna frekvencia 100KHz
- Nadprúdová ochrana
- Voľba Fast/Slow decay módu



Obr. 3-17 STMicroelectronics L6208 [32]

3.5.3 Trinamic TMC2130

Výhodou čipu TMC2130 od spoločnosti Trinamic je ponuka rôznych integrovaných funkcionalít, ktoré môže užívateľ konfigurovať prostredníctvom zbernice SPI. Tá zabezpečuje obojsmernú komunikáciu medzi čipom a príslušným mikrokontrolérom. Riadenie krokového motora je zabezpečené pomocou dvojice signálov STEP/DIR. Riadený KM podobne ako u predošlého čipu vykoná jeden krok v reakcii na nábežnú hranu STEP signálu. Výhodou tohto čipu je okrem iného programovateľná časť sekvenceru, ktorý slúži na dosiahnutie rozlíšenia až 256 mikrokrokov.[33]



Obr. 3-18 Schéma vnútornej štruktúry radiča TMC2130 [33]

TMC2130 umožňuje užívateľovi nastavenie niekoľko patentovaných funkcionalít. Keďže sú bežné krokové motory často riadené bez polohovej spätnej väzby, môže prichádzať pri ich preťažení ku strate krokov, čo vedie ku strate informácie o aktuálnej polohe. Technológia StallGuard poskytuje možnosť internej spätnej väzby záťažného uhlu krokového motora. Pri prekročení nastaviteľnej maximálnej hodnoty, je vyslaný signál do mikroprocesoru, ktorý zastaví generovania riadiaceho Step signálu, čo vedie ku okamžitému ukončeniu pohybu KM. Citlivosť technológie StallGuard disponuje 1024 úrovňami, preto je možné jej nastavenie podľa potrieb danej aplikácie.[33]

3.6 Doplnkové senzory na riadenie pohonov

3.6.1 Enkodéry

Hlavnou výhodou krokových motorov v porovnaní s ostatnými typmi pohonov, je možnosť ich riadenia bez spätnej väzby, tzv. „Open-loop control“. Hoci je tento cenovo dostupný spôsob riadenia často využívaný, problém nastáva v momente, keď je pohon preťažený a príde ku strate kroku. Z tohto dôvodu sú v niektorých aplikáciách krokové motory doplnené enkodérom, ktorého úlohou je zabezpečiť spätnú väzbu polohy, vďaka čomu môžu byť prípadné odchýlky kompenzované.

Na trhu existuje veľké množstvo enkodérov, ktoré môžeme podľa základného princípu funkcie rozdeliť na lineárne a rotačné. Oba typy enkodérov, môžu byť absolútne alebo inkrementálne. Výhodou využitia absolútnych enkodérov je možnosť absencie referenčnej kalibrácie osi po zapnutí zariadenia, keďže absolútny enkodér na rozdiel od inkrementálneho disponuje informáciou o absolútnej polohe. Ďalej je možné enkodéry rozdeliť na magnetické, optické a kapacitné. Bližším popisom funkcie jednotlivých enkodérov je uvedený v [34]. Jednoduché enkodéry komunikujú s mikrokontrolérom pomocou digitálnych pulzov, zatiaľ čo u pokročilejších sa stretáme s komunikáciou po zbernici SPI, UART alebo I2C. Moderné sofistikované absolútne enkodéry disponujúce prídavnými funkcionalitami, ako je napríklad autodiagnostika, automatická kalibrácia, prípadne nastavenia vlastných interných registrov. Ako príklad je možné uviesť absolútny enkodér Orbis od spoločnosti RLS [35]

3.6.2 Koncové senzory

Koncové senzory sú elektro-mechanické zariadenia slúžiace pri regulácii pohybu. Vo väčšine aplikácii slúžia na signalizáciu potreby zastavenia alebo zmeny smeru pohybu pohonu. Časté je taktiež ich využitie pri referencovaní osi, kedy je nulová poloha definovaná polohou koncového senzoru. Podľa princípu funkcie ich môžeme rozdeliť do troch základných kategórii.

Mechanické koncové senzory sú najjednoduchším typom koncových senzorov. Skladajú sa z tenkého plechového prúžku a malého tlačidla. Pri dotyku prichádza ku stlačeniu tlačidla, čím je odoslaný pulz do nadradeného riadiaceho systému.[36]

Optické koncové senzory sú využívané zriedkavo, predovšetkým z dôvodu vyššej ceny a komplikovanejšieho prevedenia v porovnaní s mechanickými senzormi. Princíp ich funkcie spočíva v prerušení svetelného lúču, vďaka čomu sú vhodné na bezkontaktné aplikácie. Eliminácia mechanického kontaktu taktiež pozitívne vplyva na životnosť tohto typu senzoru.[36]

Koncové senzory s Hallovým čidlom sa skladajú z dvoch hlavných častí. Magnetický senzor pracujúci na princípe Hallového efektu, poskytuje funkciu spínaču, ktorý sa zopne v prípade výskytu magnetického poľa. Druhou časťou je magnet, ktorý je z pravidla pripevnený na posuvnej časti systému označovanej ako jazdec. Výhodou tohto typu je v porovnaní s optickým senzorom možnosť kalibrácie spínacej vzdialenosti. Taktiež sa vyznačujú dobrou opakovateľnosťou detekcie.[36]

4 Návrh pohonnej jednotky

Návrh pohonnej jednotky úzko súvisí so samotným konštrukčným riešením konkrétnej polohovacej platformy. Konštrukčné riešenie však nepatrí medzi ciele tejto diplomovej práce. Podkapitola 4.1 obsahuje preto iba koncepčné riešenie polohovacej platformy. Na základe tohto koncepčného riešenia je v podkapitole 4.2 uvedená metóda redukcie s cieľom odvodenia vzťahu pre odhad potrebného krútiaceho momentu motoru. Podkapitola 4.3 následne popisuje jednotlivé komponenty testovacej pohonnej jednotky, ktoré boli zvolené za účelom verifikácie funkčnosti riadiacej jednotky, riadiaceho firmwaru a užívateľského softwaru.

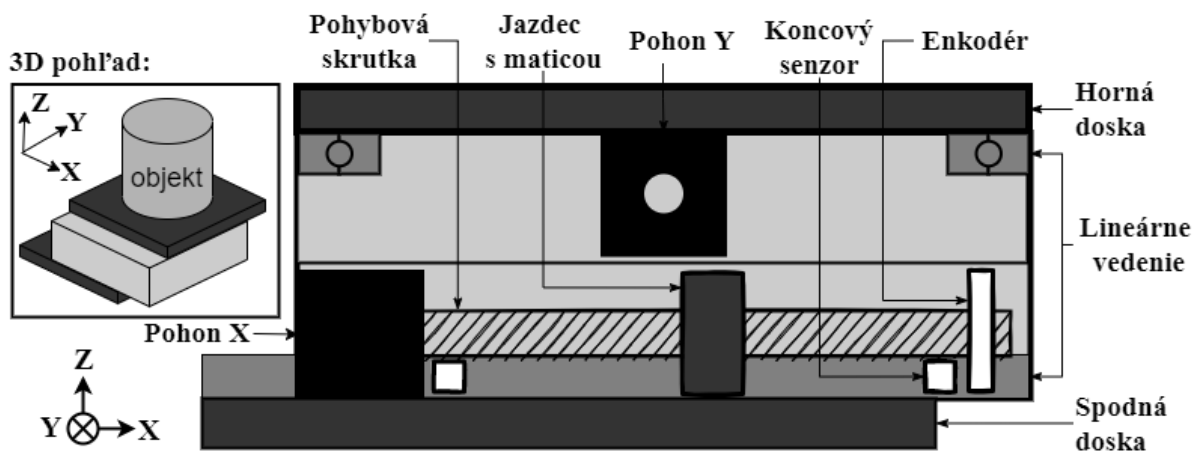
4.1 Koncepčné riešenie polohovacej platformy

Na základe prehľadu komerčne dostupných polohovacích platforiem uvedenej v podkapitole 3.2 a analýze využívaných pohonov bolo rozhodnuté, že vhodným typom pohonu je kombinácia krokového motoru a pohybovej skrutky. Toto riešenie sa ukázalo ako najvhodnejšie vďaka svojej univerzálnosti, relatívne jednoduchej montáži, cene, spoľahlivosti a možnosti digitálneho riadenia.

Prenos krútiaceho momentu motoru na pohybovú skrutku môže byť v závislosti od konštrukčného riešenia zabezpečený prevodovkou. Keďže je jednou z požiadaviek kompaktnosť navrhovaného systému, vhodné by bolo uvažovať prepojenie pomocou spojky. Možným riešením je taktiež využiť typ krokového motoru u ktorého je pohybová skrutka súčasťou jeho rotoru, vďaka čomu je eliminovaná nutnosť riešenia ich prepojenia. Toto uloženie je využité pri koncepčnom návrhu uvedenom na Obr. 4-1.

Podľa dĺžky závitovej tyče by bolo vhodné využiť radiálne ložisko na konci pohybovej skrutky. Na Obr. 4-1 je uvedené najjednoduchšie konštrukčné riešenie, ktoré uvažuje voľný koniec tyče. Toto riešenie je vhodné pre platformy s krátkym, prípadne stredne dlhým rozsahom pohybu pracujúcich na nízkych rýchlostiach.

V prípade požiadavky na riadenie s polohovou spätnou väzbou je možné využiť enkodér. Jeho uloženie je ovplyvnené priestorovými možnosťami vo vnútri platformy. Možné je jeho pripojenie na zdanú stranu krokového motoru, alebo na koniec pohybovej skrutky tak, ako je uvedené na Obr. 4-1. Súčasťou pohonnej jednotky je taktiež dvojica koncových senzorov.



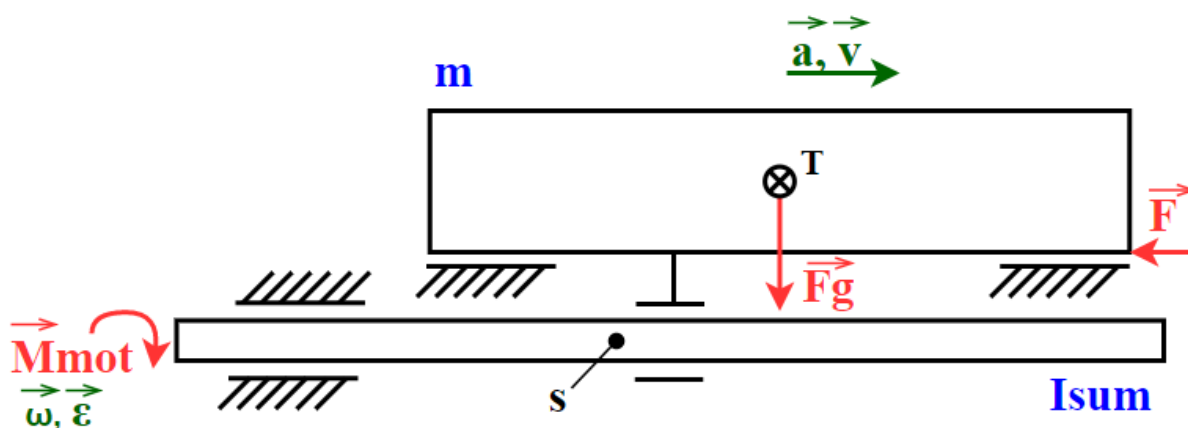
Obr. 4-1 Schéma koncepčného riešenia polohovacej platformy

4.2 Požadovaný krútiaci moment

Na zaťaženie pohonu pochopiteľne vplýva hmotnosť polohovaného objektu a požadované zrýchlenie polohovania. Zo schémy konceptu polohovacej platformy uvedenej na Obr. 4-1 vyplýva, že na záťaž pohonu v ose X vplýva taktiež hmotnosť samotnej polohovacej platformy, preto je pri výpočtoch nutné uvažovať pohon pre túto osu.

Na výpočet potrebného momentu krokového motoru je možné využiť zjednodušenie pohonnej jednotky na sústavu uvedenú na Obr. 4-2. Na túto sústavu je následne využitá metóda redukcie za účelom výpočtu potrebného momentu motoru.

Rovnica pre potrebný krútiaci moment motoru (4-6) je upravená do tvaru, v ktorom sú považované veličiny zrýchlenie, stúpanie pohybovej skrutky, celková hmotnosť (platformy a polohovaného objektu) a celkový moment zotrvačnosti (pohybovej skrutky a rotora motoru) za známe parametre.



Obr. 4-2 Sústava telies

Vzťah medzi rýchlosťou motoru a translačnou rýchlosťou objektu je možné vyjadriť rovnicou:

$$\omega = \frac{v}{s} 2\pi \quad (4-1)$$

Kde :

ω – rotačná rýchlosť motoru [rad/s]

v – translačná rýchlosť objektu [m/s]

s – stúpanie závitů pohybovej skrutky [m/otáčka]

Pre zrýchlenie motoru a polohovaného objektu platí:

$$\epsilon = \frac{a}{s} 2\pi \quad (4-2)$$

Kde :

ϵ – rotačná rýchlosť motoru [rad/s²]

a – horizontálne zrýchlenie objektu [m/s²]

Pri redukcii na translačný pohyb je možné získať redukovanú hmotnosť sústavy zo vzťahu:

$$\frac{1}{2} m_{red} v^2 = \frac{1}{2} m v^2 + \frac{1}{2} I_{sum} \omega^2$$

$$m_{red} = m + I_{sum} \frac{4\pi^2}{s^2}$$
(4-3)

Kde :

m_{red} – redukovaná hmotnosť sústavy [kg]

v – translačná rýchlosť objektu [m/s]

m – celková hmotnosť (platforma + objekt manipulácie) [kg]

I_{sum} – celkový moment zotrvačnosti (pohybová skrutka + rotor motora) [kg.m²]

ω – uhlová rýchlosť motora [rad/s]

s – stúpanie závitú pohybovej skrutky [m/otáčka]

Redukovanú silu je možné získať z rovnosti výkonu v redukovanej a riešenej sústave:

$$P_{red} = P_{sustavy}$$

$$F_{red} v = M \omega - F v$$

$$F_{red} = \frac{M}{s} 2\pi - g m f_1$$
(4-4)

Kde :

M – moment motora [Nm]

F – trecia sila v lineárnom vedení [N]

g – tiažové zrýchlenie [m/s²]

f_1 – súčiniteľ trenia vo vedení [-]

Pre zrýchlenie translačného pohybu následne platí:

$$a = \frac{F_{red}}{m_{red}}$$

$$a = \frac{\frac{M}{s} 2\pi - g m f_1}{m + I_{sum} \frac{4\pi^2}{s^2}}$$
(4-5)

Úpravou rovnice (4-5) je možné určiť požadovaný krútiaci moment¹ ako:

$$M = \frac{s m (a + g f_1)}{2\pi} + \frac{a I_{sum} 2\pi}{s}$$
(4-6)

¹ Uvedená rovnica slúži na hrubý odhad potrebného krútiaceho momentu. Pri konkrétnom konštrukčnom riešení by bolo potrebné okrem iného započítať účinnosti jednotlivých komponentov (závitová tyč, lineárne vedenie...)

4.3 Výber komponentov

Pohonná jednotka sa skladá z krokového motoru, jeho radiča, koncových senzorov a taktiež je možné využiť enkodér. Keďže neexistuje konštrukčné riešenie platformy, komponenty nie sú dimenzované na konkrétne parametre. Ich výber slúži primárne na zostavenie testovacej pohonnej jednotky, slúžiacej na verifikáciu funkcionality riadiacej jednotky, ktorej návrhom sa zaoberá Ondryáš [39]. Taktiež bude táto testovacia pohonná jednotka ďalej využitá na overenie riadiaceho algoritmu a užívateľského softwaru, ktorých návrh je hlavný cieľ tejto práce.

4.3.1 Krokový motor

Pri výbere krokového motoru bolo potrebné eliminovať konštrukčné problémy spojené s uložením závitovej tyče a jej prepojením s motorom. Z toho dôvodu bol zvolený typ krokového motoru využívajúci pohybovú skrutku, ktorá je priamo súčasťou jeho rotora. V prípade realizácie konštrukčného riešenia by bolo vhodné uvažovať nad guľôčkovou pohybovou skrutkou s cieľom dosiahnutia vyššej presnosti polohovania. Na verifikáciu funkcionality systému je možné považovať skrutku s trapézovým závitom za dostatočnú. Prípadné využitie spojky by viedlo ku rozšíreniu možností pri voľbe konkrétneho motoru, keďže by nebolo potrebné vyberať z typov motorov integrujúcich pohybovú skrutku tak, ako je uvedené na Obr. 4-3.



Obr. 4-3 Krokový motor s integrovanou pohybovou skrutkou [37]

Parametre zvoleného krokového motoru sú uvedené v Tab. 4-1.

Tab. 4-1 Parametre zvoleného krokového motoru [37]

Obecné parametre		Elektrické parametre	
Uhol kroku	$1,8^\circ \pm 5\%$	Nominálne napätie	3,06 V
Počet fázy	2	Nominálny prúd	1,7 A
Statický moment	560 mN.m	Odpor vinutia	$1,8 \Omega \pm 10\%$
Zbytkový moment	24 mN.m	Indukčnosť vinutia	$3,8 \text{ mH} \pm 20\%$
Zotrvačnosť rotoru	82 g.cm^2		
Hmotnosť	0,36 kg		

4.3.2 Radič motoru TMC2130

Na riadenie krokového motoru bol zvolený radič TMC2130. Medzi základné funkcionality tohto radiča patrí digitálne STEP/DIR riadenie, integrovaná regulácia prúdu, až 256 mikrokrokov na plný krok. Pokročilejšie funkcie a jeho schéma sú uvedené na konci podkapitoly 3.5. Tento radič je možné zakúpiť ako vývojový komponent uvedený na Obr. 5-1, vďaka čomu je možná jeho jednoduchá integrácia do vývojovej jednotky, tak ako je popísané v podkapitole 5.1. Samostatný integrovaný obvod, bol využitý pri návrhu riadiacej jednotky v práci Ondryáša [39]. Umiestnenie čipu na navrhnutej riadiacej doske je možné vidieť na Obr. 5-2.

4.3.3 Enkodér a koncové senzory

Hoci je výhodou krokových motorov riadenie bez spätnej väzby, jednou z požiadaviek bolo navrhnuť systém, ktorý disponuje možnosťou odstránenia potreby inicializačného referencovania osi po zapnutí zariadenia. Z tohto dôvodu bol zvolený absolútny enkodér Orbis od spoločnosti RLS. Výrobca u tohto typu enkodéru uvádza 14-bitové rozlíšenie, možnosť autokalibrácie po montáži a integrovanú autodiagnostiku. Na komunikáciu s mikrokontrolérom využíva rozhranie SPI. Tento senzor sa skladá z čítacej hlavy a magnetického krúžku, ktoré sú uvedené na Obr. 4-4.



Obr. 4-4 Absolútny enkodér RLS Orbis [35]

Na základe prehľadu koncových senzorovo uvedených v podkapitole 3.6.2, sa riešenie využívajúce Hallov senzor a magnet na jazdci pohybovej skrutky ukázalo ako najvhodnejšie. Výrobu koncového senzoru na Obr. 4-5 zabezpečil v rámci svojej práce Ondryáš [39].

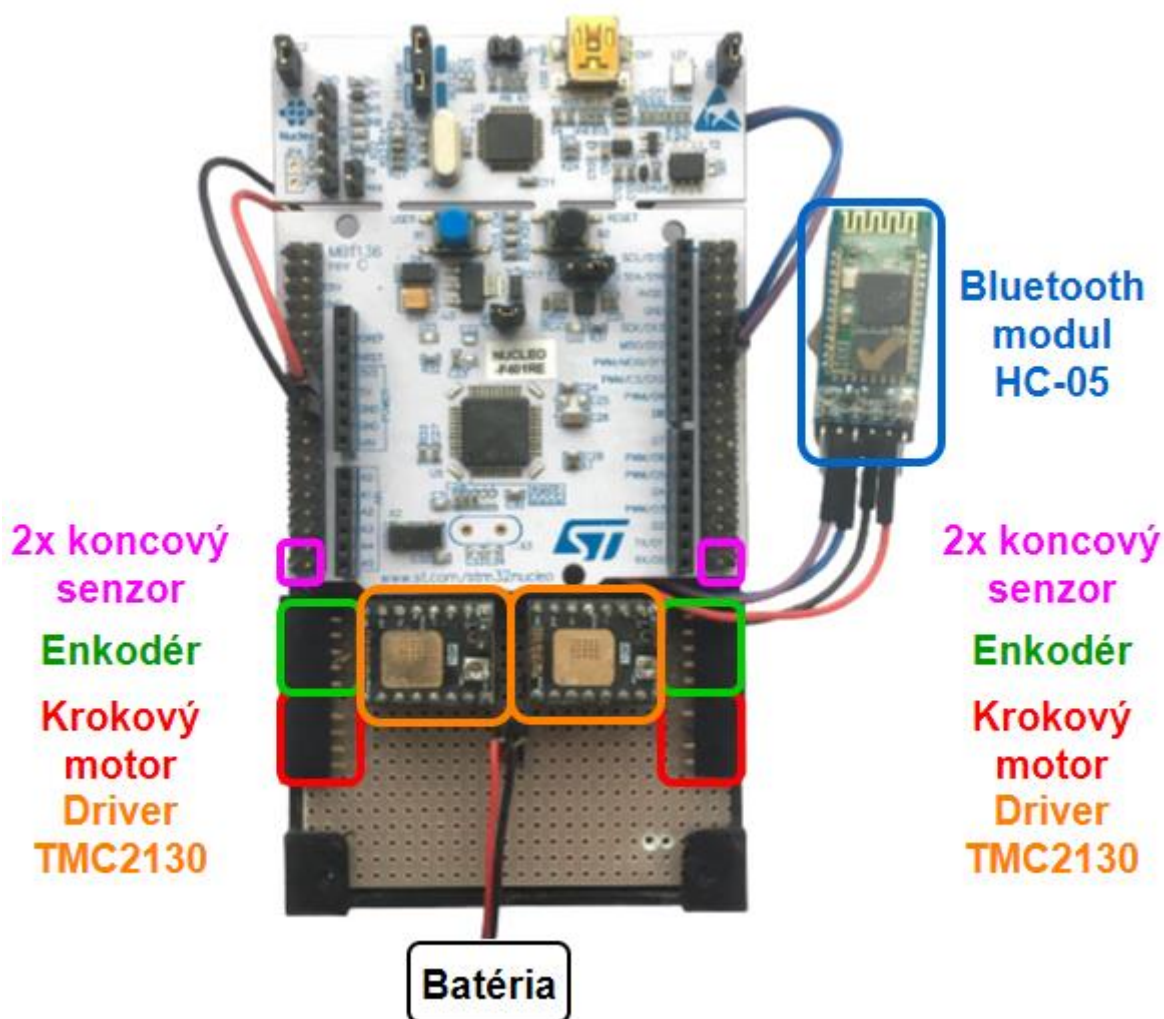


Obr. 4-5 Koncový senzor s Hallovým čidlom [39]

5 Riadiaca jednotka

5.1 Vývojová jednotka

Ako uvádza Ondryáš [39], riadiaca jednotka, ktorej návrh je cieľom jeho práce, využíva mikrokontrolér STM32-L476. Za účelom efektívnejšieho plnenia úloh definovaných v podkapitole 2.1, bola zakúpená vývojová doska Nucleo-F401RE od STMicroelectronics, na ktorú som v rámci tejto práce vytvoril jednoduchý shield, ktorý umožňuje pripojenie základných komponentov, ako je uvedené na Obr. 5-1. Vďaka tomu mohol byť riadiaci firmware, vyvíjaný paralelne s vývojom riadiacej jednotky. Prototypová riadiaca jednotka uvedená v 5.2 však obsahuje komponenty, ktoré neboli dostupné ako vývojové moduly². Vývoj firmwaru mohol byť preto dokončený a testovaný až po vytvorení funkčného prototypového riešenia riadiacej jednotky.

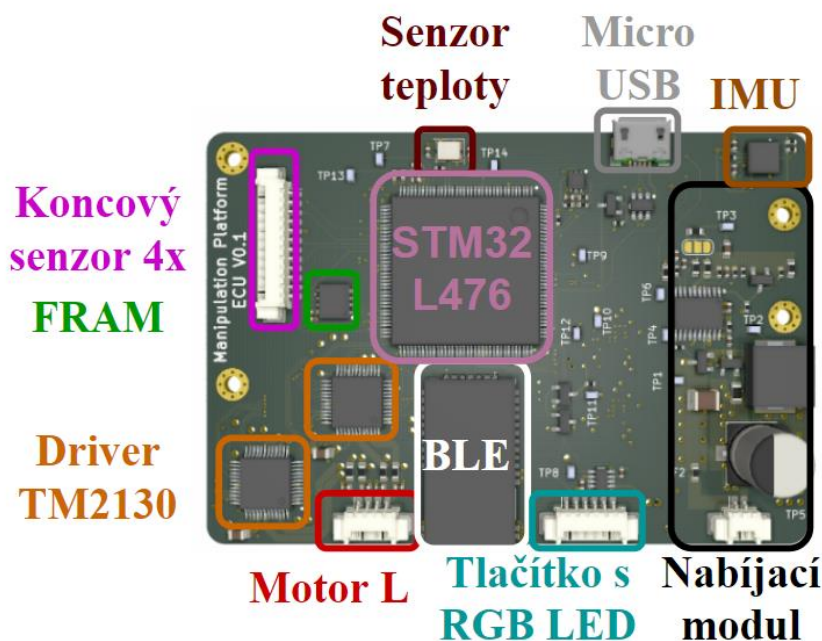


Obr. 5-1 Vývojová riadiaca jednotka

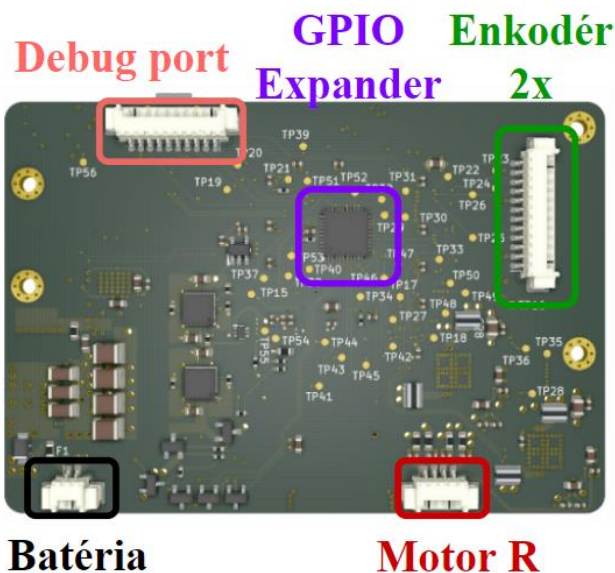
² Vývojový modul môže byť označovaný skratkou BOB odvodenou do anglického ekvivalentu „Breakout Board“

5.2 Prototypové riešenie riadiacej jednotky

Riadiaca doska, ktorú vytvoril v rámci svojej práce Ondryáš [39] je obojstranne osadená komponentami. Z Obr. 5-2 a Obr. 5-3 je evidentné, že sa jedná o výrazne komplexnejší systém v porovnaní s vývojovým zapojením, ktoré slúžilo iba na testovanie základných funkcionalít. Nasledujúce kapitoly sa preto zaoberajú návrhom komunikačného protokolu, firmwaru a softwaru pre túto komplexnú riadiacu jednotku, integrujúcu jednotlivé pridané komponenty.



Obr. 5-2 Finálne riešenie riadiacej jednotky (pohľad zhora) [39]



Obr. 5-3 Finálne riešenie riadiacej jednotky (pohľad zdola) [39]

6 Návrh komunikačného protokolu

Bezdrôtová komunikácia medzi PC softwarom a riadiacou jednotkou je jedným z hlavných benefitov navrhovaného riešenia. Prepojenie týchto dvoch systémov je zabezpečené prostredníctvom technológie Bluetooth Low Energy (BLE), ktorej sa detailnejšie venuje vo svojej práci Ondryš [39]. BLE modul, ktorý je súčasťou riadiacej jednotky komunikuje s mikrokontrolérom jednotky prostredníctvom zbernice UART.

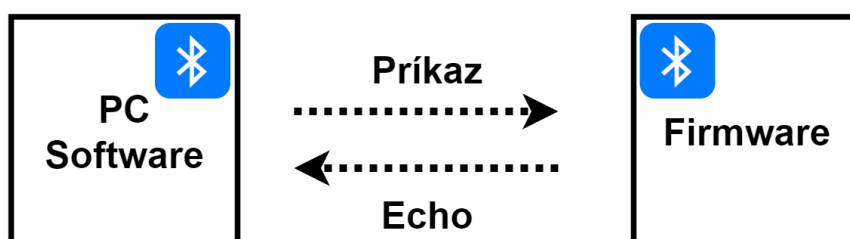
Komunikácia je založená na princípe obojsmernej výmeny reťazcov znakov medzi PC a riadiacou jednotkou. Podľa schémy na Obr. 6-1 je príkazovou správou možné označiť reťazec znakov odoslaných z PC softwaru. Z riadiacej jednotky je odosielaný reťazec znakov označovaný ako Echo správa. Keďže prichádza ku zmene obsahu oboch týchto správ v závislosti na type príkazu, stave zariadenia a jednotlivých parametrov, je potrebný návrh komunikačného protokolu.

Hlavnou úlohou komunikačného protokolu je definovať pravidlá skladby jednotlivých reťazcov znakov s cieľom zabezpečiť jasne definovanú tvorbu a spracovanie správ vedúce ku správne riadeniu systému.

6.1 Požiadavky na komunikačný protokol

Skladba jednotlivých správ je spojená s funkcionalitami riadiacej jednotky, ktoré v rámci svojej práce uvádza Ondryš [39]. Pri návrhu komunikačného protokolu treba taktiež dbať na požiadavku univerzálnosti a možnosti rozširovania o ďalšie funkcionality. Z toho dôvodu sú kladené na komunikačný protokol nasledujúce požiadavky:

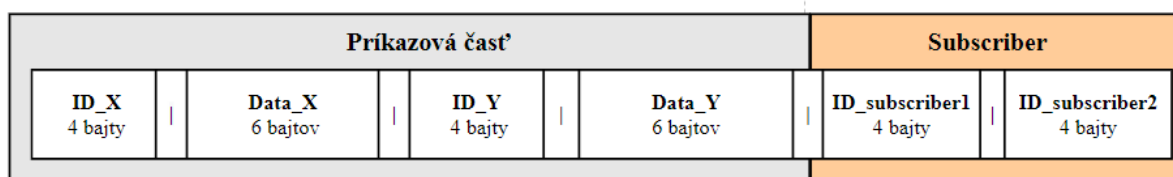
- Možnosť riadiť obe osi (X,Y) polohovacej platformy súčasne
- Schopnosť prenosu dáta zo senzorov do PC softwaru
- Komunikácia s komponentami riadiacej jednotky (nastavovanie ich registrov)
- Univerzálnosť (jednoduché pridanie ďalších funkcionalít)
- Pevne daná dĺžka dátového rámcu s možnosťou rozšírenia (jednoduchšie spracovanie)
- Možnosť vyčítavania stavu/hodnoty užívateľom zvolených parametrov



Obr. 6-1 Typy správ v bezdrôtovej komunikácii

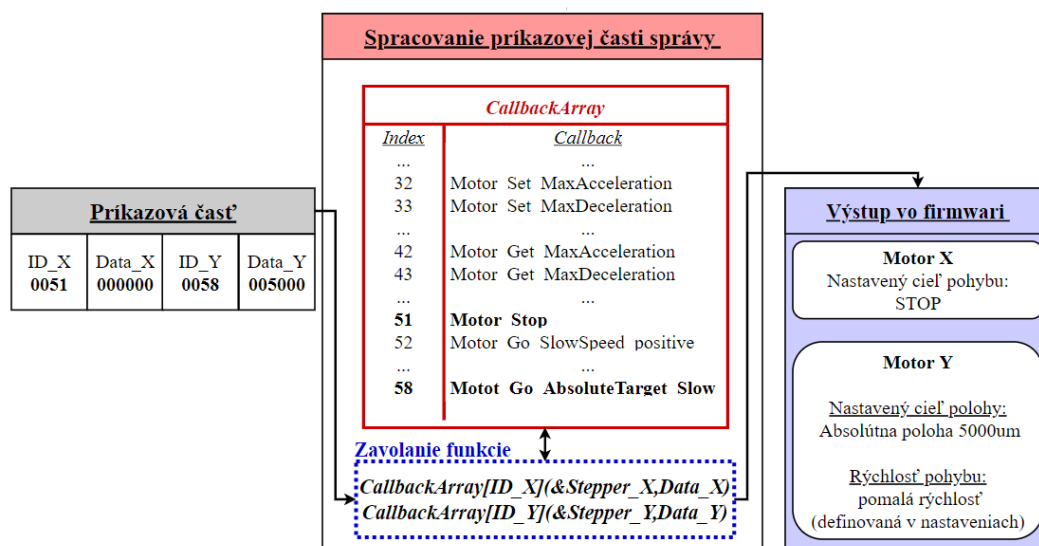
6.2 Príkazová správa

Úlohou príkazovej správy je prenos užívateľských povelov zo softwaru do firmwaru riadiacej jednotky. Odosielanie správ z užívateľského softwaru nastáva s periódou 100 ms. Pri návrhu komunikačného protokolu bol zvolený prístup, ktorý priradzuje jednotlivým príkazom unikátne identifikátory (ID). Výhodou tohto prístupu je značné zjednodušenie, sprehľadnenie a taktiež možnosť jednoduchého rozširovania komunikačného protokolu. Ako je vidieť na Obr. 6-2, príkazová časť správy sa skladá sa z dvojice ID a Data pre smer X a Y, vďaka čomu je možné riadiť obe osi platformy súčasne. Ak príkaz nepotrebuje dáta, sú odosielané tzv. DummyData³. Správa je taktiež rozšírená o časť Subscriber, ktorému sa bližšie venuje podkapitola 6.3.



Obr. 6-2 Zloženie príkazovej správy

Princíp vyhodnotenia príkazovej správy vo firmware riadiacej jednotky je založený na usporiadaní adries príkazových funkcií do vektora. Konkrétne ID je následne možné chápať ako pozíciu (index) v danom vektore, na ktorej sa nachádza adresa príslušnej funkcie (označovanej ako „Callback“). Toto usporiadanie prináša výhodu v podobe univerzálnosti volania funkcie pomocou prijatého ID v príkazovej časti správy a taktiež možnosti jednoduchého rozširovania počas ďalšieho vývoja. Ako príklad je možné uviesť príkaz na zastavenie pohonu v ose X a súčasne nastavenie absolútnej cieľovej polohy pohonu v ose Y. Adresy týchto funkcií sú uložené vo vektore *CallbackArray* na pozíciách definovaných ich ID. Na Obr. 6-3 je uvedená jednoduchá schéma spracovania tejto časti správy.



Obr. 6-3 Základný princíp vyhodnocovania identifikátorov príkazov

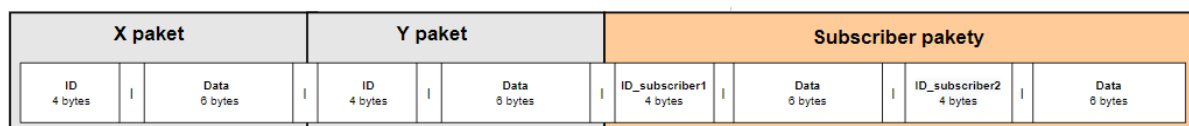
³ DummyData je označenie pre reťazec „000000“. Príklad je možné vidieť na Obr. XX. Účelom tohto reťazcu je zachovať pevnú dĺžku príkazovej správy.

6.3 Subscriber

Úlohou tejto časti príkazovej správy je poskytnúť užívateľovi možnosť definovať požiadavku na vyčítanie hodnoty konkrétnej premennej. Týmto premennými môžu byť napríklad stav batérie, teplota, prípadne rôzne ďalšie dáta. Princíp funkcie je podobný ako v predchádzajúcej kapitole. Užívateľ si zvolí z ponuky identifikátor premennej (označovaný ako „ID_subscriber“), ktorej hodnotu chce vyčítať. Vo firmware riadiacej jednotky je tento identifikátor spracovaný a riadiaca jednotka odošle požadované dáta v echo správe ako súčasť Subscriber paketu, ktorý je možné vidieť na Obr. 6-4. Príklad tohto paketu je uvedený v nasledujúcej podkapitole.

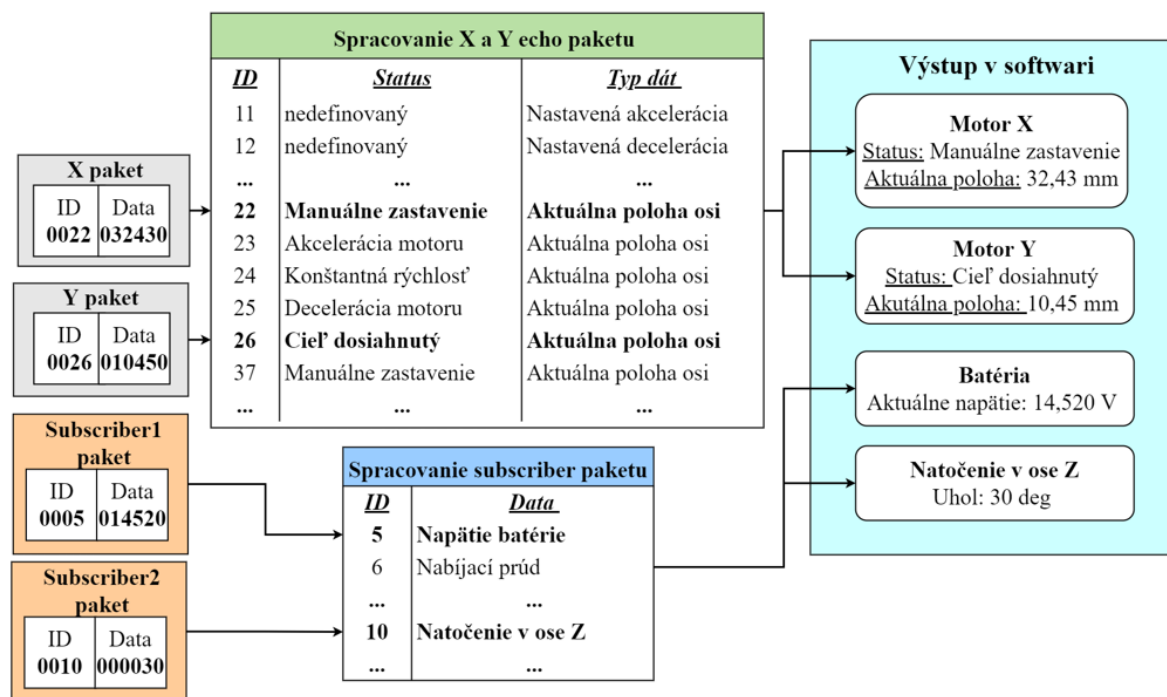
6.4 Echo správa

Echo správa je odosielaná z riadiacej jednotky do užívateľského PC softwaru, v ktorom je následne spracovaná. Zloženie správy na Obr. 6-4 je podobné ako u príkazovej správy. Jednotlivé echo ID X a Y paketov plnia dve základné úlohy. Prvou úlohou je definovať typ echo dát za účelom správneho spracovania v softwari. Druhou úlohou je definovať status riadených komponentov s cieľom informovať užívateľa o aktuálnom stave zariadenia. Subscriber pakety sa skladajú z kópie identifikátoru subscriberu z príkazovej správy doplneného o príslušné dáta.



Obr. 6-4 Zloženie echo správy

Príklad Echo správy a jej výstup v užívateľskom softwari sú schematicky popísané na Obr. 6-5.



Obr. 6-5 Základný princíp vyhodnocovania echo identifikátorov

7 Návrh riadiaceho algoritmu (firmware)

Podľa stanovených cieľov uvedených v podkapitole 2.1 je návrh riadiaceho algoritmu považovaný za hlavný cieľ tejto diplomovej práce. Nasledujúce podkapitoly sa preto zaoberajú stanovením požiadaviek na algoritmus, jeho zložením, princípu vytvárania knižníc pre komponenty a taktiež popisom vetiev algoritmu obsahujúcich doplnkové algoritmy na obsluhu jednotlivých komponentov.

7.1 Požiadavky na riadiaci algoritmus

Podobne ako na komunikačný protokol uvedený v kapitole 6, je aj na návrh riadiaceho algoritmu kladených niekoľko požiadaviek. Cieľom je navrhnúť firmware pre riadiacu jednotku navrhnutú v rámci práce Ondryáša [39]. Požiadavky na algoritmus preto vyplývajú zo systémových požiadaviek, ktoré stanovil v rámci svojej práce. Z toho dôvodu je potrebné navrhnúť algoritmus tak, aby bol schopný využívať a riadiť jednotlivé zvolené komponenty. Dôležité je pri návrhu algoritmu taktiež myslieť na jeho jednoduché rozširovanie.

Na základe systémových požiadaviek a komponentov zvolených pri návrhu riadiacej jednotky, je možné pre riadiaci algoritmus definovať nasledujúce požiadavky:

- Bluetooth komunikácia s užívateľským softwarom
- Riadenie krokových motorov vo dvoch osiach
- Komunikácia s komponentami pohonnej jednotky(enkodéry, radiče, koncové senzory)
- Riadenie napájania (odmeriavanie napätia batérie a nabíjacieho prúdu, detekcia pripojenia nabíjačky, spánkový režim – vypnutie vybraných periférií)
- Vyhodnocovanie dát z pridaných senzorov: teplotné čidlo, gyroskop, akcelerometer
- Zápis parametrov do FRAM pamäte
- Zakomponovanie GPIO expandéru do riadiaceho systému
- Integrácia užívateľského tlačidla s RGB LED signalizáciou
- Jednoduché rozširovanie, rozdelenie do viacerých zdrojových súborov a pod.

7.2 Knižnice na obsluhu komponentov

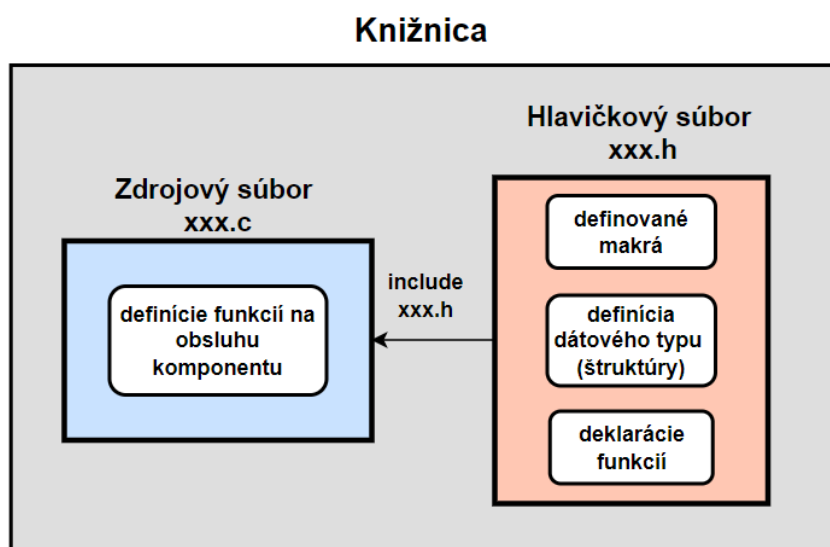
Za účelom udržania prehľadnosti algoritmu, bol pri jeho návrhu zvolený nasledujúci prístup. Jednotlivým komponentom riadiacej jednotky boli vytvorené vlastné firmwarové knižnice. Ako je uvedené v schéme na Obr. 7-1, knižnica sa skladá zo zdrojového a hlavičkového súboru.

Zdrojový súbor slúži na definíciu funkcií slúžiacich na obsluhu komponentu, pre ktorý je knižnica vytvorená. Tieto funkcie sú následne využívané v riadiacom algoritme.

Úlohu hlavičkového súboru je možné rozdeliť na tri časti. V prvej časti sú definované makrá, ktoré sú využívané za účelom definície konštánt. V niektorých prípadoch môžu makrá taktiež vykonávať jednoduché pomocné funkcie.

Pre príslušný komponent je v hlavičkovom súbore vytvorená definícia dátového typu. Vďaka tomu je možné deklarovať v rámci riadiaceho algoritmu pre každý komponent špecifickú premennú. Touto premennou je dátová štruktúra, s ktorou následne pracujú jednotlivé funkcie algoritmu. Bližším zložením tejto štruktúry sa zaoberá podkapitola 7.2.1.

Poslednou časťou je deklarácia prototypov funkcií. Po prepojení hlavičkového súboru s iným zdrojovým súborom, ktorým môže byť napríklad *main.c*, je možné vďaka tomu z tohto zdrojového súboru zavolať funkcie definované v rámci vytvorenej knižnice.



Obr. 7-1 Obecná schéma vytváraných knižníc

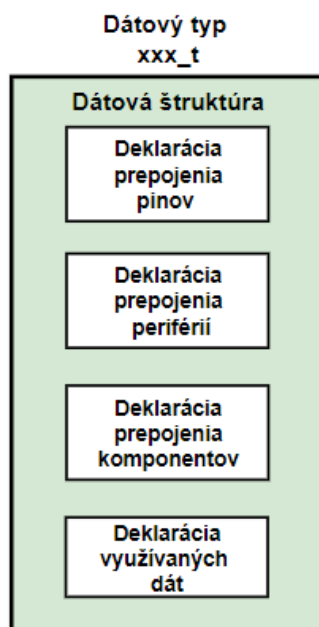
7.2.1 Spôsob vytvárania dátových štruktúr

Nestála situácie na trhu elektrokomponentov má za následok nedostatok čipov. Vhodné je preto pri návrhu dátových štruktúr pre jednotlivé komponenty dbať na jednoduché firmwarové prispôsobenie v prípade, ak by prišlo ku zmene mikrokontroléru, alebo odstráneniu niektorého z komponentov riadiacej jednotky. Za týmto účelom sú v rámci vytvorených dátových štruktúr vyhradené premenné, ktoré slúžia na definíciu vzájomného prepojenia komponentu s príslušnými digitálnymi IO pinmi mikrokontroléru. Riadiace funkcie následne pristupujú ku jednotlivým pinom pomocou týchto premenných. Vďaka tomu je možné pri zmene mikrokontroléru jednoduché prispôsobenie sa prípadným zmenám v hardwarovom prepojení, čo vedie ku zvýšeniu univerzálnosti vytvorených knižníc. Univerzálnosť bola úspešne testovaná pri prechode z vývojovej jednotky na prototypovú riadiacu jednotku, ktoré využívajú rozdielne mikrokontroléry. Bližšie sú tieto riadiace jednotky popísané v kapitole 5.

Na riadenie a komunikáciu s jednotlivými komponentami sú využívané rôzne periférie mikrokontroléru. Aby bolo možné s týmito perifériami pracovať v rámci vytvorených funkcií, je potrebné definovať v navrhovaných dátových štruktúrach ukazovatele. Týmto ukazovateľom sú následne priradené adresy preddefinovaných dátových štruktúr slúžiacich na obsluhu periférií mikrokontroléru.

Riadiaca jednotka je navrhnutá tak, že niektoré komponenty sú vzájomne prepojené. Napríklad radič krokového motora TMC2130 je prepojený s GPIO expandérom. V rámci štruktúry sú preto za účelom definície tohto prepojenia vytvorené ukazovatele na dátové štruktúry ostatných pripojených komponentov.

Každý komponent pracuje s dátami, či už ide o ich odosielanie, prijímanie alebo využitie pri riadení. Z toho dôvodu je v rámci dátovej štruktúry na Obr. 7-2 potrebné vyhraďiť pre tieto premenné pamäťové miesto.



Obr. 7-2 Zloženie dátovej štruktúry na obsluhu komponentu

7.2.2 Príklad konkrétnej dátovej štruktúry

Premenná, ktorá je v riadiacom algoritme využívaná pri obsluhu radiču krokového motora je deklarovaná ako dátový typ *TMC2130_t*. Hardwarové prepojenie radiča s ostatnými komponentami riadiacej dosky vychádza z jej návrhu, ktorý uvádza vo svojej práci Ondryáš [39].

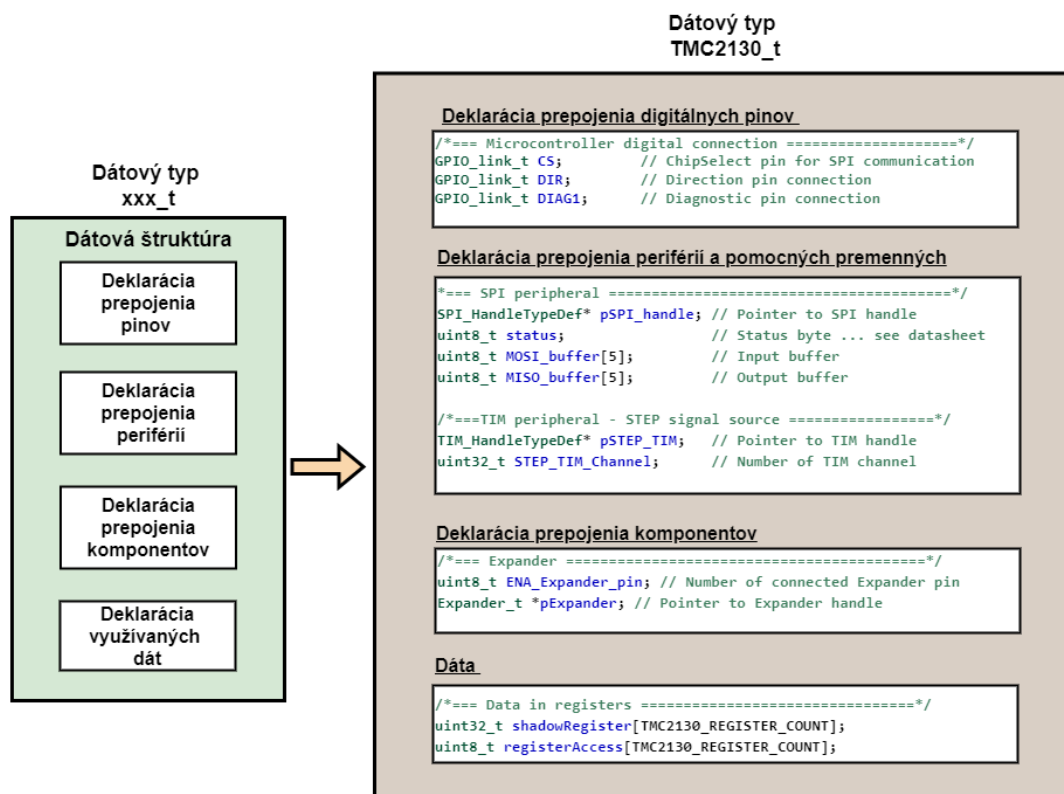
Digitálne vstupné a výstupné piny radiča (CS,DIR,DIAG1) sú pripojené priamo k pinom mikrokontroléra. V rámci dátovej štruktúry sú preto deklarované premenné typu *GPIO_link_t*, ktorých úlohou je definovať pripojenie ku konkrétnemu pinu mikrokontroléra.

Komunikácia radiča s mikrokontrolérom je zabezpečená pomocou SPI. Z toho dôvodu je potrebné vytvorenie ukazovateľa na preddefinovanú štruktúru využívanú pri obsluhu SPI. Taktiež je potrebné deklarovať príslušné zásobníky (MOSI,MISO), ktorých veľkosť je definovaná datasheetom radiča TMC2130 [33].

Radič krokového motora je ovládaný digitálnymi pulzami (STEP). Na ich generovanie je využitý jeden z časovačov mikrokontroléra. Za účelom možnosti zmeny parametrov tohto časovača je preto potrebné vytvorenie ukazovateľa na preddefinovanú štruktúru využívanú pri obsluhu periférie TIM. Taktiež je potrebné vyhradiť pamäťové miesto na definíciu využívaného kanálu časovača.

Na aktiváciu radiča slúži pin ENA. Prototypová riadiaca jednotka využíva na jeho ovládanie GPIO expandér. Z toho dôvodu je potrebné vytvorenie ukazovateľa na jeho dátovú štruktúru.

Vektory *shadowRegister* a *registerAccess* obsahujú dáta využívané pri nastavovaní registrov radiča.



Obr. 7-3 Príklad zloženia dátovej štruktúry na obsluhu TMC2130

7.2.3 Zoznam vytvorených knižníc

Za účelom lepšej orientácie v zdrojových a hlavičkových súboroch, Tab. 7-1 uvádza vzájomné súvislosti medzi požadovanými funkcionalitami, využívanými komponentami a jednotlivými súbormi firmwaru, ktoré boli v rámci tejto práce vytvorené.

Tab. 7-1 Zoznam vytvorených knižníc na obsluhu komponentov

Funkcionalita	Komponent	Zdrojový súbor	Hlavičkový súbor	Dátový typ
Riadenie motorov	Driver TMC2130	TMC_2130.c	TMC_2130_MaskShift.h	-
			TMC_2130_Registers.h	-
			TMC_2130.h	TMC2130_t
	Absolútny enkodér	Encoder.c	Encoder.h	Encoder_t
	Koncový senzor	Endstop.c	Endstop.h	Endstop_t
	Krokový motor	Stepper.c	Stepper.h	Stepper_t
Bezdrôtová komunikácia	Bluetooth Low Energy modul	BLE_module.c	BLE_module.h	BLE_module_t
		Command.c	Command.h	Command_t
		Echo.c	Echo.h	Echo_t
Rozšírenie I/O pinov	GPIO Expander	Expander.c	Expander.h	Expander_t
Ukladanie dát a parametrov	FRAM pamäť	FRAM.c	FRAM.h	FRAM_t
Power management	Výkonový modul (batéria, nabíjačka, tranzistory)	Power_module.c	Power_module.h	Power_module_t
Užívateľské tlačidlo s RGB LED	Push button controller + 3 x MOSFET pre LED	User_button.c	User_button.h	User_button_t
Odmeriavanie teploty riadiacej dosky	Senzor teploty	Temp_sensor.c	Temp_sensor.h	Temp_sensor_t
Odmeriavanie natočenia	Inertial measurement unit (IMU)	IMU.c	IMU.h	IMU_t

7.3 Štruktúra riadiaceho algoritmu

Riadiaci algoritmus, ktorý je súčasťou hlavného súboru *main.c* je možné koncepčne rozdeliť do štyroch častí zobrazených na Obr. 7-4.

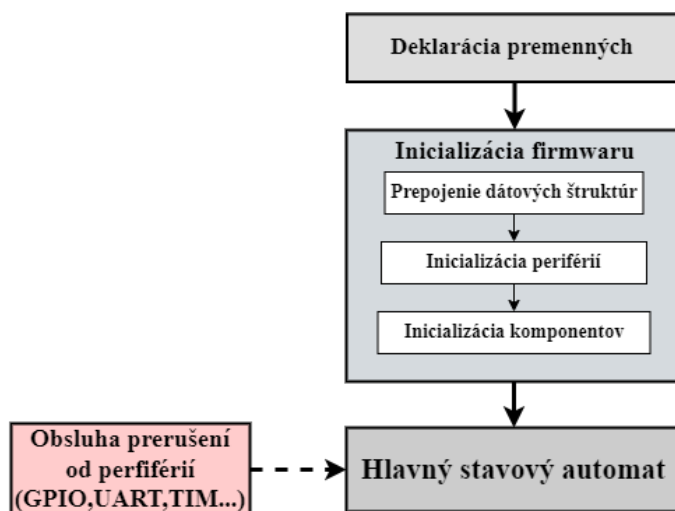
Navrhnuté dátové typy sú použité na deklaráciu premenných. Vďaka tomu je možné v rámci firmwaru pristupovať ku dátovým štruktúram komponentov a perifériám mikrokontroléru.

Inicializácia firmwaru sa skladá z troch častí:

- Prepojenie dátových štruktúr je zabezpečené pomocou ukazovateľov na ich adresy v pamäti. Popisom tohto prístupu sa zaoberá podkapitola 7.2.1.
- Inicializácia periférií slúži na nastavenie jednotlivých periférií mikrokontroléru, ktoré využíva na komunikáciu s pripojenými komponentami. Konkrétnu hodnotu parametrov je možné definované v prostredí STM CubeMX, ktorý slúži na vygenerovanie preddefinovaných dátových štruktúr na riadenie periférií.
- Inicializácia komponentov je potrebná na nastavenie registrov radiča krokového motora, počiatočnú akvizíciu dát zo senzorov a zahájenie bezdrôtovej komunikácie

Po inicializácii komponentov prechádza riadiaci algoritmus do hlavného stavového automatu, v rámci ktorého je systém ďalej riadený. Jeho konkrétnemu popisu sa venuje podkapitola 7.5.

Za účelom zefektívnenia algoritmu je možné vybrané periférie nastaviť do módu, kedy neblokujú mikrokontrolér ale pracujú v prerušení. Až po splnení definovanej podmienky prichádza k zavolaniu niektorej z funkcií, ktorá je často označovaná ako „Callback“.

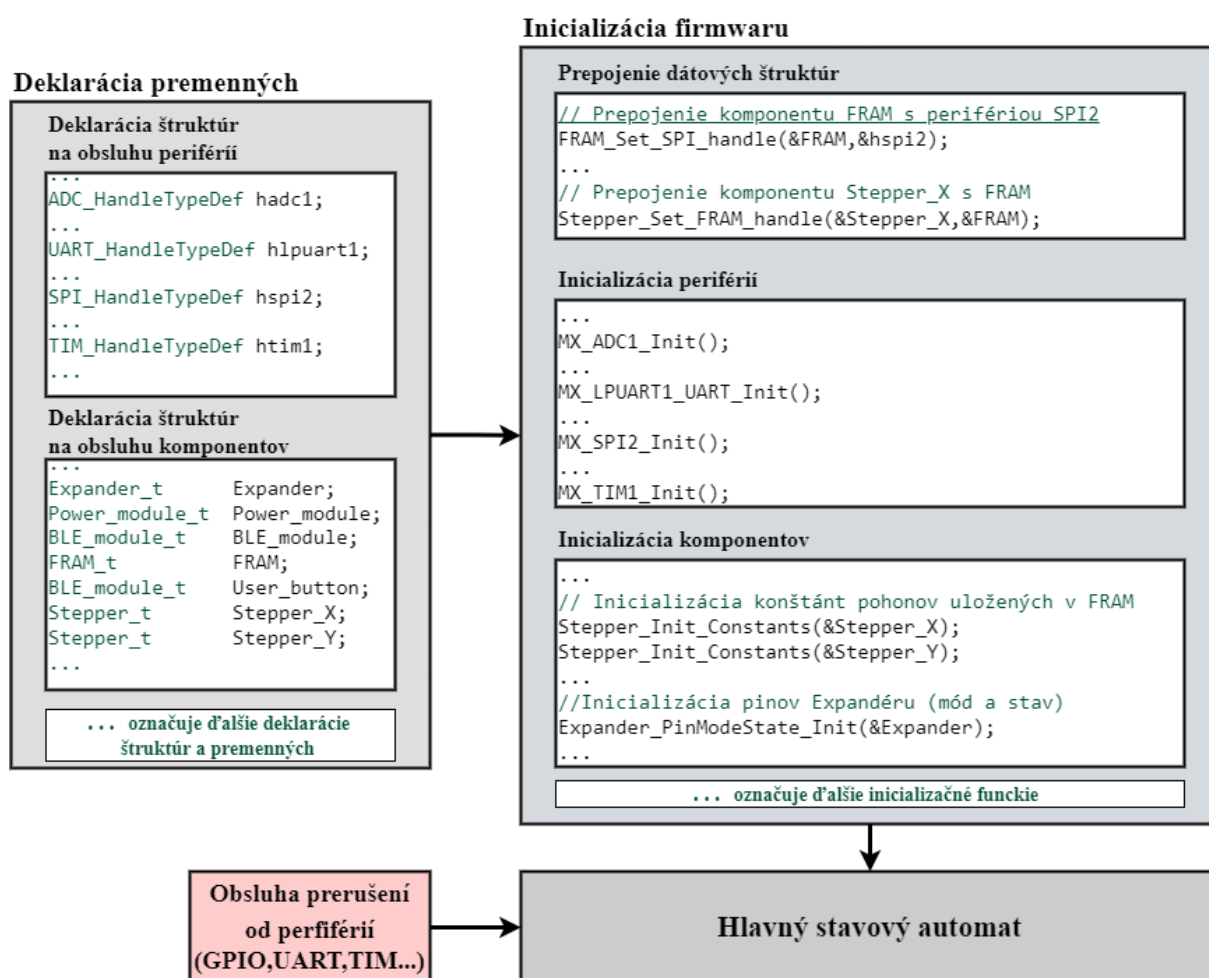


Obr. 7-4 Schéma štruktúry riadiaceho algoritmu

7.4 Deklarácia premenných a inicializácia firmwaru

Po spustení firmwaru prichádza ku deklarácií premenných pomocou dátových typov definovaných v jednotlivých knižniciach. Tým sú alokované pamäťové miesta pre dátové štruktúry slúžiace na obsluhu periférií a miesta pre dátové štruktúry slúžiace na prácu s komponentami.

V inicializačnej časti je potrebné prepojiť dátové štruktúry komponentov spolu so štruktúrami príslušných periférií tak, ako uvádza Obr. 7-5. Taktiež je u niektorých komponentov potrebné vzájomné prepojenie ich dátových štruktúr. Inicializácia periférií ma za úlohu nastaviť parametre zvolených periférie ktoré sú definované vo vývojovom prostredí STM CubeMX. Inicializované periférie sú následne využité na komunikáciu s komponentami za účelom nastavenia ich interných registrov, prípadne vyčítanie parametrov. Taktiež v inicializačnej časti prichádza k povoleniu prerušení a úvodnej akvizícii dát zo senzorov. Inicializácia firmwaru je ukončená LED sekvenciou, ktorá užívateľovi signalizuje pripravenosť zariadenia. Riadiaci algoritmus následne prechádza do hlavného stavového automatu.



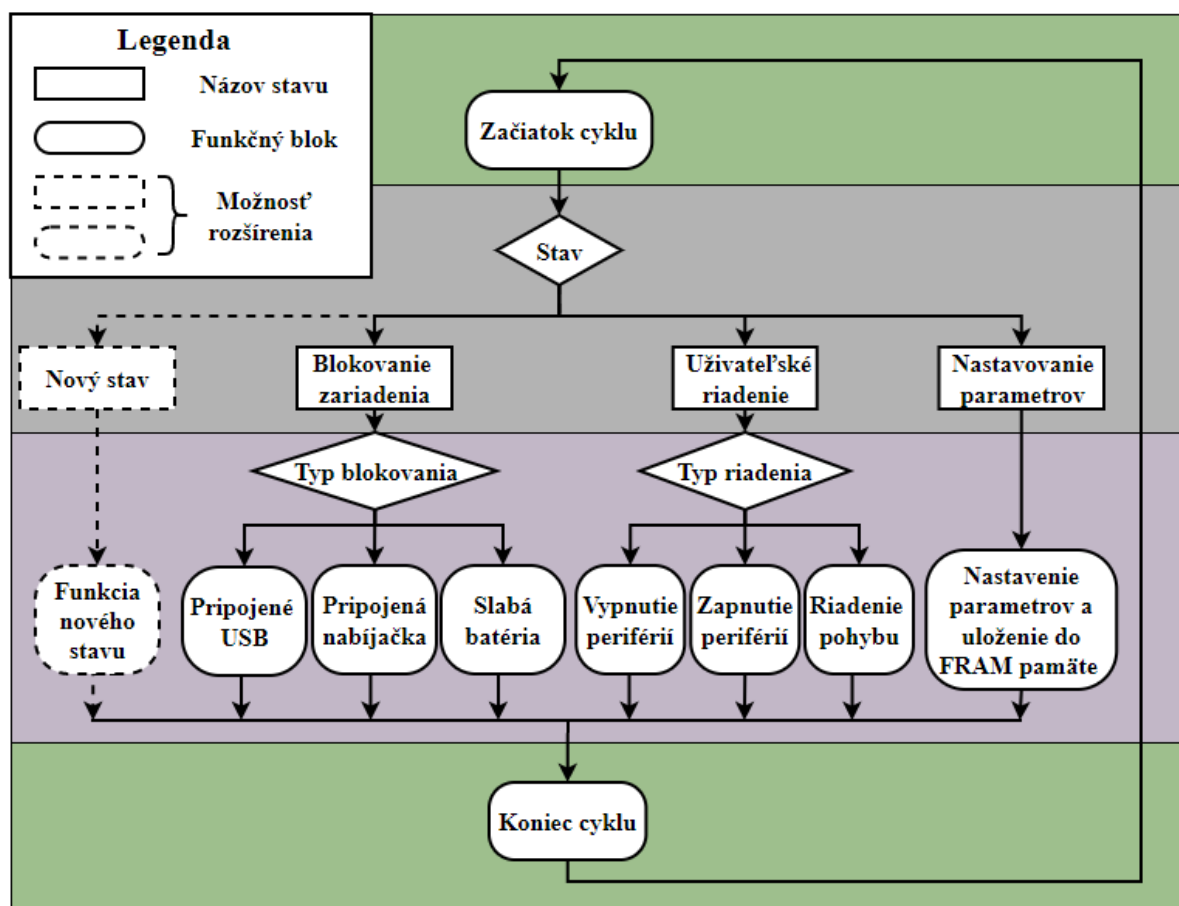
Obr. 7-5 Príklad deklarácie premenných a inicializácie firmwaru

7.5 Hlavný stavový automat

Prístup k riadeniu pomocou stavového automatu bol zvolený z dôvodu jeho prehľadnosti a možnosti jednoduchého rozširovania pomocou pridávania ďalších vetiev tak, ako je možné vidieť na Obr. 7-6 vľavo. Stavový automat je súčasťou súboru *main.c*, kde beží v jeho nekonečnej *while* slučke.

Schému hlavného stavového automatu je možné rozdeliť do dvoch úrovní. V prvej úrovni prichádza ku nastaveniu hlavného stavu riadiaceho automatu. Tento stav je ovplyvnený zložením príkazovej správy, stavom jednotlivých komponentov a hodnotami interných parametrov. Stav je taktiež ovplyvňovaný obsluhou prerušení.

Automat môže nadobúdať tri hlavné stavy, ktorými sú blokovanie zariadenia, užívateľské riadenie a nastavovanie parametrov. V závislosti od hlavného stavu môže prísť k ďalšiemu vetveniu stavu do druhej úrovne podľa jeho typu. V tejto úrovni prichádza k obsluhu jednotlivých funkčných blokov, v rámci ktorých sú riadené jednotlivé komponenty. Cyklus môže byť ukončený odoslaním echo správy do riadiaceho softwaru. Samotné zloženie echo správy a jej odosielanie na konci cyklu je ovplyvnené aktuálnymi hodnotami interných parametrov riadiaceho algoritmu. Bližším popisom jednotlivých stavov a mechanizmov funkčných blokov sa zaoberajú nasledujúce podkapitoly.



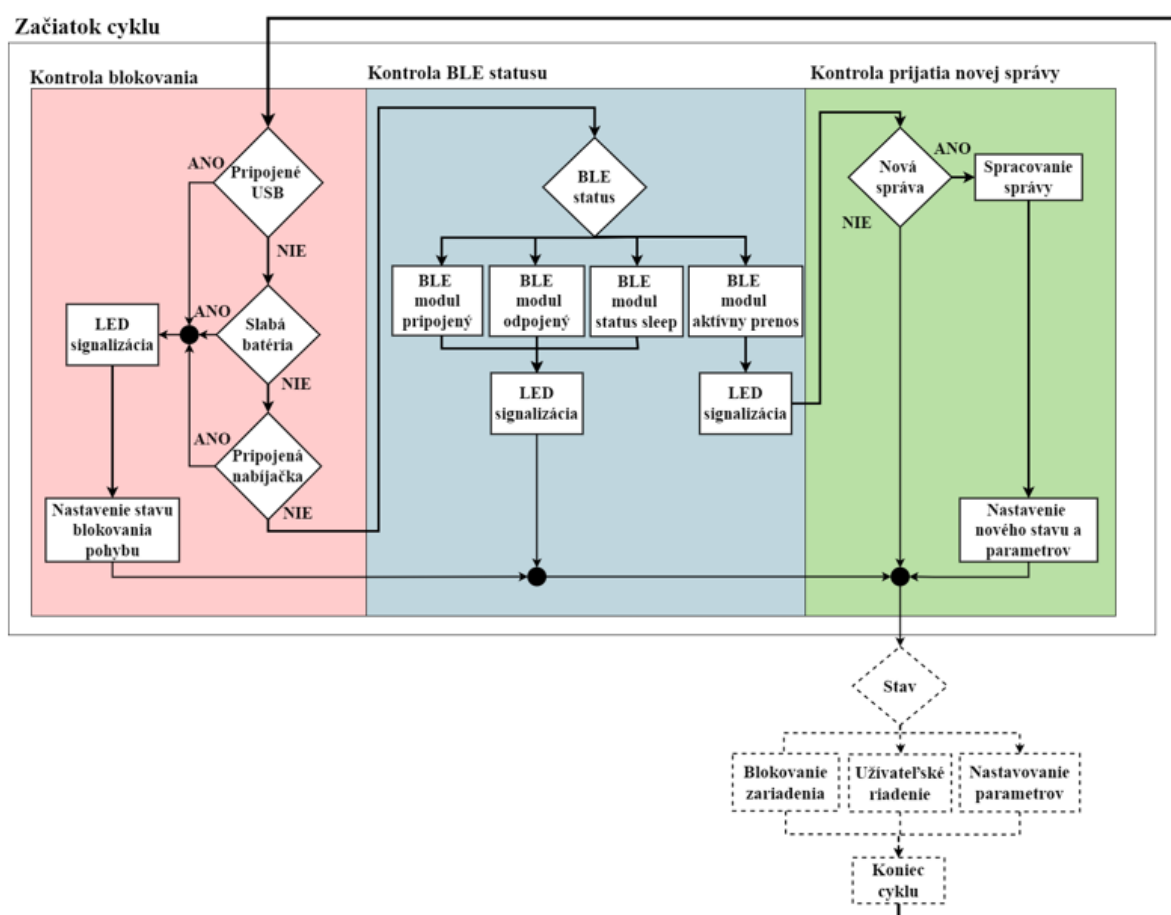
Obr. 7-6 Schéma hlavného stavového automatu

7.5.1 Začiatok cyklu stavového automatu

V úvode cyklu stavového automatu je potrebná kontrola stavu jednotlivých komponentov tak, ako je uvedené na Obr. 7-7. Na začiatku je potrebná kontrola stavu komponentov na základe ktorého môže nastať blokovanie zariadenia. Je preto nutná kontrola signalizácie pripojeného USB, kontrola signalizácie slabej batérie a prípadného pripojenia nabíjačky.

V prípade, že nepríde k detekcii potreby blokovania zariadenia, prichádza ku kontrole stavu BLE modulu. Jednotlivé typy statusu sú bližšie popísané v podkapitole 7.6, ktorá popisuje zakomponovanie modulu do riadiaceho algoritmu. Ak je aktivovaný prenos dát BLE modulu a detekovaná nová prijatá správa, prichádza ku jej spracovaniu a nastavovaniu nového stavu automatu a príslušných riadiacich parametrov.

Za účelom signalizácie stavu zariadenia je riadiaca jednotka vybavená tlačidlom s RGB LED krúžkom. Vďaka tomu môže byť užívateľovi poskytnutá jednoduchá hardwarová spätná väzba prostredníctvom rôznych LED sekvencií. Popisu užívateľského tlačidla a jeho funkcionalitám sa venuje podkapitola 7.9.5.



Obr. 7-7 Schéma procesov na začiatku cyklu stavového automatu

7.5.2 Blokovanie zariadenia

Tento stav je nastavený internými procesmi riadiaceho algoritmu. Užívateľ preto nevie ovplyvňovať zmenu tohto stavu zmenami príkazovej správy. Na možnosť prechodu do iného stavu je potrebné odstrániť príčinu blokácie, ktorými môžu byť pripojené USB, pripojená nabíjačka, alebo batéria vybitá pod definovanú kritickú úroveň. Keďže je v uvedených prípadoch nutné zabrániť pohybu zariadenia, je v každom z nich okrem iného potrebné zabezpečiť blokovanie motorov. Ako uvádza Ondryáš [39], pri testovaní riadiacej jednotky bolo zistené chybné pripojenie USB. Na prechod do tohto stavu blokovania je preto využité manuálne nastavovanie kontrolného bitu. Toto nastavovanie bude po dodaní funkčnej riadiacej jednotky nahradené príslušnou funkciou.

7.5.3 Užívateľské riadenie

Ako vyplýva z názvu, tento stav slúži na riadenie platformy pomocou užívateľských príkazov. Podľa druhu príkazu sa tento stav ďalej vetví na:

- **Vypnutie periférií** – možnosť vypnutia vybraných periférií užívateľským príkazom za účelom šetrenia batérie
- **Zapnutie periférií** – pred riadením je v prípade vypnutých periférií potrebné ich opätovné zapnutie užívateľským príkazom
- **Riadenie motorov** – tejto funkcii algoritmu sa venuje samostatná podkapitola 7.7.

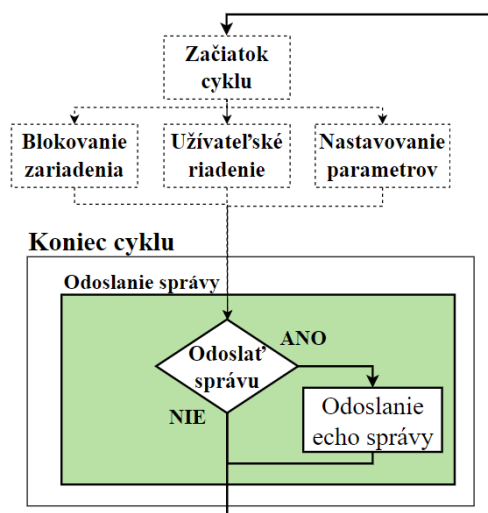
Ovládaniu stavu periférií za účelom šetrenia batérie sa bližšie venuje podkapitola 7.9.3, ktorá popisuje prácu s Výkonovým modulom.

7.5.4 Nastavovanie parametrov

Užívateľské parametre, ako napríklad hodnota akcelerácie, rýchlostí, alebo taktiež rôzne ďalšie parametre je možné uložiť do FRAM pamäte. Možnosť nastavenia výrobných a užívateľských parametrov pozitívne vplýva na univerzálnosť navrhovaného systému. Integráciu FRAM pamäte do riadiaceho algoritmu sa venuje podkapitola 7.8.

7.5.5 Koniec cyklu

V prípade, že počas predchádzajúceho cyklu stavového automatu prišlo k prijatiu príkazovej správy, prichádza na konci cyklu ku reakcii firmwaru na v podobe odoslania echo správy do užívateľského softwaru, ako je schematicky uvedené na Obr. 7-8.



Obr. 7-8 Koniec cyklu stavového automatu

7.6 Komunikácia pomocou BLE modulu

Úlohou BLE modulu je zabezpečenie bezdrôtovej komunikácie riadiacej jednotky s užívateľským PC softwarom. BLE modul osadený na riadiacej doske komunikuje s mikrokontrolérom prostredníctvom zbernice UART. Na signalizáciu svojho stavu modul využíva dvojicu digitálnych pinov, ktoré sú na riadiacej doske pripojené ku GPIO expandéru. Význam jednotlivých kombinácií stavu pinov je uvedený v Tab. 7-2.

Tab. 7-2 Význam indikátorov statusu BLE modulu

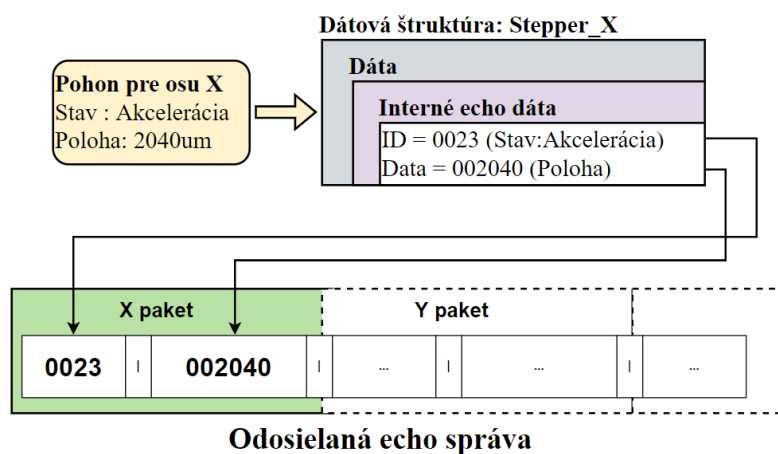
STATUS1 IND	STATUS2 IND	Status	Popis
1	1	Shutdown mode	Modul prechádza do tzv. deep-sleep módu
1	0	Standby mode	Modul vysiela tzv. advertising pakety a čaká na pripojenie.
0	0	BLE connected	Modul je spárovaný s PC
0	1	Data Session open	Modul je pripojený a komunikuje

7.6.1 Prijímanie príkazových správ

Veľkosť prijatej príkazovej správy je definovaná pravidlami navrhnutého komunikačného protokolu, ktorého popisu sa venuje podkapitola 6.2. Aby neprichádzalo k blokovaniu stavového automatu, je UART obsluhovaný v prerušení. Po prijatí požadovaného počtu znakov je zavolaná funkcia, ktorá nastaví signalizáciu novej prijatej správy. V prípade, že sú dodržané všetky potrebné podmienky, prichádza na začiatku cyklu hlavného stavového automatu ku spracovaniu správy podľa princípu uvedeného v 6.2. Výsledkom spracovania je nastavenie nového stavu automatu a riadiacich parametrov príslušných komponentov.

7.6.2 Tvorba odosielaných echo správ

Podobne ako u príkazovej správy, je aj u echo správy jej veľkosť definovaná pravidlami komunikačného protokolu. Obr. 7-9 uvádza princíp vytvorenia časti echo správy pomocou interných echo dát komponentu. V príklade je uvedený stav riadenia pohonu v ose X.



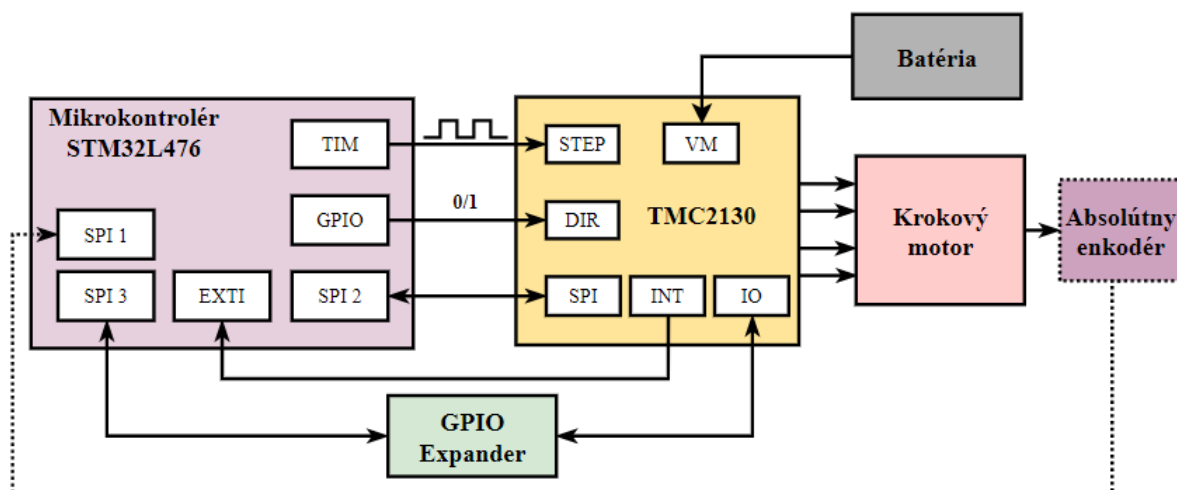
Obr. 7-9 Princíp vytvorenia časti echo správy

7.7 Riadenie krokových motorov

Ako je uvedené na Obr. 7-10, krokový motor je možné riadiť pomocou dvojice digitálnych signálov STEP/DIR. Na generovanie STEP signálu slúži jeden z časovačov periférie TIM nastavený v móde generovania výstupného PWM signálu. Jednotlivé kroky motoru sú výsledkom reakcie radiča na nábežné hrany STEP signálu. Smer otáčania je riadený pomocou digitálneho signálu DIR. Radič TMC2130 obsahuje interné registre, ktoré slúžia na parametrizáciu jeho jednotlivých funkcionalít. Nastavovanie týchto registrov prostredníctvom zbernice SPI je popísané v podkapitole 7.7.1.

Jednou z požiadaviek pri návrhu riadiaceho algoritmu bolo zakomponovanie absolútneho enkodéru. Výhodou absolútneho enkodéru je, že pozná svoju absolútnu polohu ihneď po zapnutí. Vďaka tomu systém po spustení pozná aktuálnu polohu v oboch osiach, čo umožňuje odstránenie potreby referencovanie osi vždy po zapnutí zariadenia. Hlavnou výhodou krokových motorov je však možnosť ich riadenia bez polohovej spätnej väzby. Keďže je potrebné dbať na univerzálnosť riešenia, algoritmus je navrhnutý tak, aby bolo možné enkodér odstrániť. Touto problematikou sa bližšie zaoberá podkapitola 7.7.2.

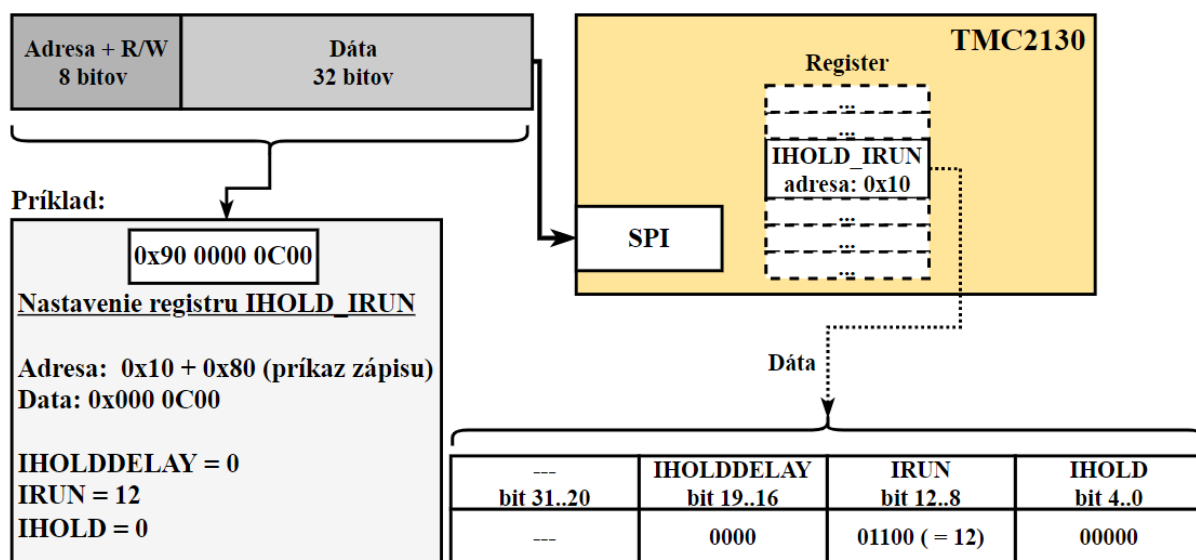
Navrhnutý algoritmus poskytuje užívateľovi možnosť ovládať krokové motory v módoch manuálneho, absolútneho a relatívneho polohovania. Zatiaľ čo u manuálneho riadenia je cieľom dosiahnuť požadovanú rýchlosť a smer pohybu, účelom absolútneho polohovania je automatické dosiahnutie užívateľom definovanej polohy. V prípade relatívneho pohybu prichádza ku pohybu po definovaných inkrementoch. Užívateľ si na polohovanie môže vybrať dva stupne rýchlosti. Hodnoty rýchlostí a taktiež akcelerácie je podľa potreby aplikácie možné zmeniť priamo zo softwaru ako je uvedené v kapitole 8. Pre správnu funkciu riadenia je taktiež potrebné definovať parametre ako stúpanie závitovej tyče, počet krokov motoru na otáčku, počet mikrokrokov, a rôzne ďalšie. Všetky tieto parametre spolu s parametrami rýchlostí a akcelerácie je možné uložiť do pamäte FRAM, vďaka čomu nie je po opakovanom zapnutí zariadenia potrebné ich opätovné nastavovanie. Zápis parametrov do FRAM je popísaný v podkapitole 7.8.



Obr. 7-10 Schéma riadiaceho systému pre krokový motor

7.7.1 Komunikácia s radičom TMC2130

Po zapnutí riadiacej jednotky je potrebné inicializovať jednotlivé interné registre radiča. Tie obsahujú nastavenia rôznych parametrov ako napríklad počet mikrokrokov, požadovaný prúd v motore, módy diagnostických pinov a iné. Taktiež je možné nastaviť parametre rôznych sofistikovaných funkcií ako StallGuard, CoolStep, SpreadCycle, ktoré sú bližšie popísané v [33]. Radič TMC2130 komunikuje s mikrokontrolérom po zbernici SPI prostredníctvom 40 bitových správ. Ako je vidieť na Obr. 7-11, správa sa skladá z 8 bitovej adresy registru, ku ktorej je pričítavaný tzv. R/W bajt, ktorý slúži na odlíšenie príkazu na zapísanie a vyčítanie hodnoty registru. Adresa je doplnená o 32bitov dát, ktoré sú do registru zapísané. V prípade požiadavky na vyčítavanie registru radič prijaté dáta ignoruje. TMC2130 disponuje internou prúdovou spätnou väzbou. Súčasťou schémy na Obr. 7-11 je preto príklad nastavenia registru IHOLD_IRUN, ktorý slúži na nastavenie parametrov riadenia prúdu.



Obr. 7-11 Nastavovanie registrov radiča TMC2130

Ako uvádza [33] hodnotu parametru IRUN je možné chápať ako škálovací parameter pri nastavovaní požadovaného prúdu v cievkach motoru. Výpočet konkrétnej hodnoty prúdu je daný rovnicou:

$$I_{RMS} = \frac{C_s + 1}{32} * \frac{V_{FS}}{R_{sense} + 20m\Omega} * \frac{1}{\sqrt{2}} \quad (7-1)$$

Kde:

I_{RMS} – stredná kvadratická hodnota prúdu [A]

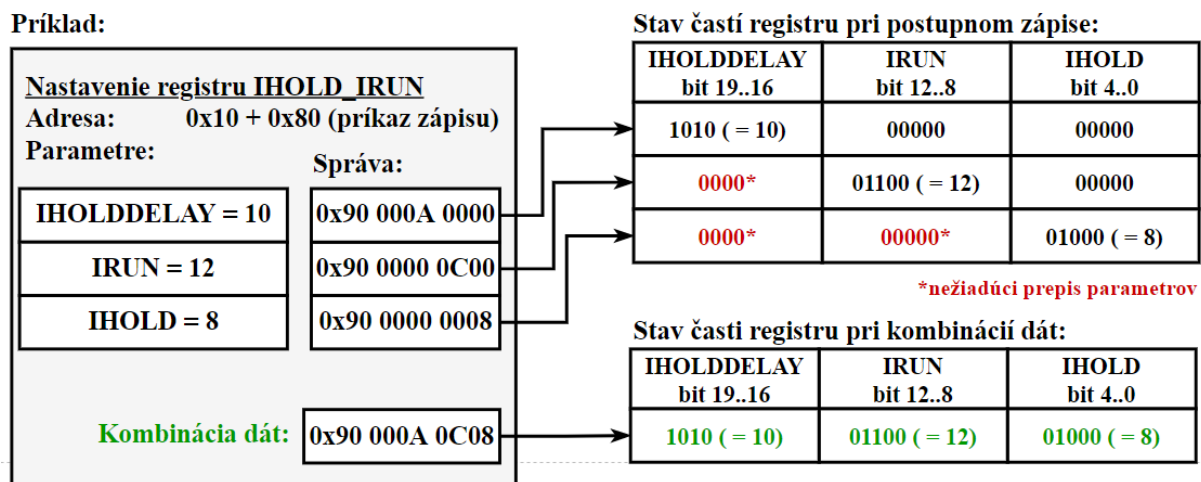
C_s – Current Scale (ekvivalent nastavenej hodnoty IRUN 0..31) [-]

V_{FS} – referenčné napätie [V]

R_{sense} – hodnota Sense rezistoru⁴ [Ω]

⁴ Rezistor R_{sense} je využívaný pri odmeriavaní prúdu v cievke ako je uvedené v podkapitole 3.4.1

Problém nastáva v momente, ak je potrebné nastaviť viacero parametrov, ktoré sú súčasťou jedného registru. Ako vyplýva z Obr. 7-12, ak by boli správy odosielané postupne, prichádzalo by ku prepisu jednotlivých častí registru. Z toho dôvodu je potrebné nastavenie jednotlivých častí registru skombinovať a odoslať ako jednu správu.



Obr. 7-12 Kombinácia dát pri zápise do registru radiča TMC2130

Kombinácia odosielaných dát do jednej správy vedie k správnomu zápisu parametrov do jednotlivých častí registru. Nevýhodou tohto prístupu je však výrazné zníženie čitateľnosti častí firmwaru, ktorá sa zaoberá inicializáciou registrov radiču krokového motora. Za účelom prehľadnosti a ľahšieho prístupu k registrom, bolo potrebné v rámci knižnice pre obsluhu TMC2130 vytvoriť funkcie, ktoré slúžia na nastavovanie jednotlivých parametrov. Výhoda v porovnaní s priamym zápisom je uvedená na Obr. 7-13.

Priamy zápis kombinovaných dát:

```
TMC2130_Transmit(&(Stepper_X.driver), 0x90000A0C08)
```

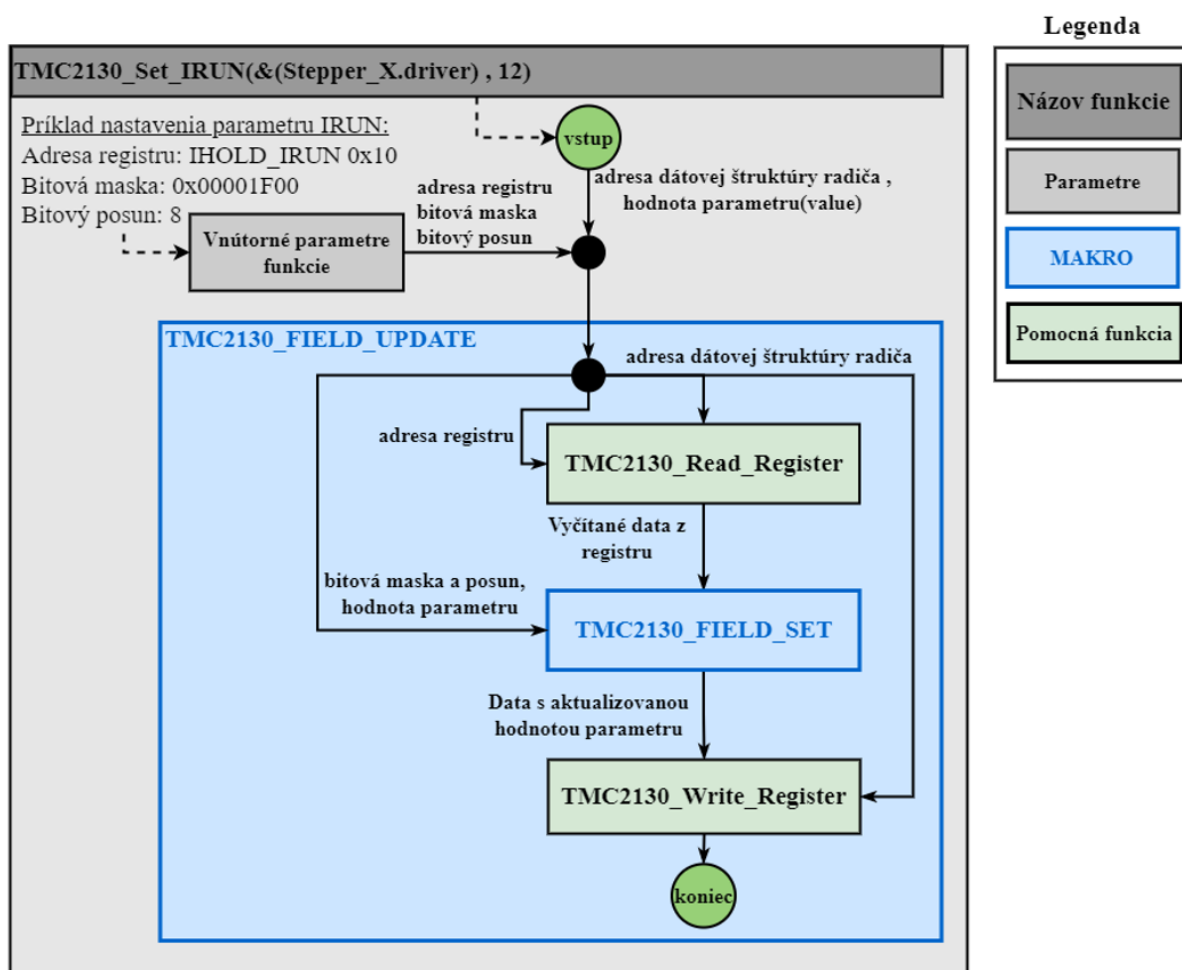
Využitie funkcií vytvorenej knižnice:

```
TMC2130_Set_IHOLDDELAY(&(Stepper_X.driver), 10);
TMC2130_Set_IRUN(&(Stepper_X.driver), 12);
TMC2130_Set_IHOLD(&(Stepper_X.driver), 8);
```

Obr. 7-13 Porovnanie výhod využitia funkcií knižnice

Jednotlivé funkcie slúžiace na nastavovanie registrov obsahujú makrá, ktoré sú definované v súbore *TMC_2130.h*. Tieto makrá využívajú SPI komunikáciu a aplikáciu bitových masiek a posunov. Tento prístup, ktorý bol inšpirovaný [38], je schematicky popísaný na Obr. 7-14. Ako príklad je využitá funkcia na nastavovanie prúdu motora.

V prvej časti prichádza k vyčítaniu dát z príslušného registru pomocou funkcie *TMC2130_Read_Register*. Následne je úlohou makra *TMC2130_FIELD_SET* správne skombinovať vyčítané dáta z registru a hodnotu nastavovaného parametru pomocou bitovej masky a posunu. V poslednej časti prichádza ku zápisu nových dát do príslušného registru radiča. Obr. 7-14 taktiež uvádza príklad vnútorných parametrov funkcie. Hodnoty týchto parametrov súvisia s usporiadaním registrov, ktoré je definované datasheetom [33] a sú špecifické pre každú funkciu. Vďaka týmto funkciám je zabezpečená prehľadnosť algoritmu a znížené riziko chyby pri úpravách algoritmu.



Obr. 7-14 Schéma princípu nastavenia registrov TMC2130 využitím funkcií

7.7.2 Odmeriavanie polohy

Obecne je informáciu o polohe v danej ose možné získať dvoma spôsobmi. V prípade riadenia bez polohovej spätnej väzby prichádza k počítaniu STEP pulzov vygenerovaných jedným z časovačov periférie TIM. Druhým riešením je využitie polohovej spätnej väzby v podobe enkodéru. V oboch prípadoch je ale informácia o polohe relatívna, keďže pri reštarte prichádza ku vynulovaniu premenných bez ohľadu na to, v akej polohe sa zariadenie nachádza.

Ako je uvedené v úvode podkapitoly 7.7, jedným z požiadaviek je preto zakomponovanie absolútneho enkodéru do riadiaceho algoritmu za účelom odstránia potreby referencovania osi po zapnutí zariadenia. Keďže je cieľom vytvoriť univerzálne riešenie, potrebné je navrhnúť riadiaci algoritmus tak, aby systém fungoval aj v platformách, ktoré nevyužívajú enkodér. Z toho dôvodu bol zvolený nasledujúci prístup.

Absolútny enkodér slúži na inicializačné odmeranie absolútnej polohy po zapnutí zariadenia. Zmeraná absolútna poloha je prevedená na absolútnu hodnotu pulzov podľa rovnice:

$$AHP = AHO * MNK * KNO + \frac{PVO * MNK * KNO}{REN} \quad (7-2)$$

Kde :

AHP - absolútna hodnota pulzov [-]

AHO - absolútna hodnota celých otáčok vyčítaná z enkodéru [-]

MNK - počet mikrokrokov na krok [-]

KNO - počet krokov na otáčku

PVO - meraná poloha v otáčke (u zvoleného enkodéru RLS Orbis - 14bitové číslo)

REN - rozlíšenie enkodéru

Získaná absolútna hodnota pulzov je zapísaná do premennej, ktorá slúži na ďalšie počítanie STEP pulzov. Následne prichádza ku klasickému riadeniu bez polohovej spätnej väzby, ktoré je popísané v úvode podkapitoly 7.7. Vďaka tomu je možné navrhnutý systém aplikovať aj do platforiem, ktoré nedisponujú enkodérom. U týchto platforiem však bude potrebné referencovanie osi vždy po zapnutí zariadenia.

Možným kompromisom medzi absolútnym enkodérom a potrebou inicializačného referencovania osi je využitie FRAM pamäte. Počas vypínacej sekvencie systému je možné uložiť absolútnu hodnotu počtu STEP pulzov. Vďaka tomu môže byť inicializačné odmeriavanie enkodérom nahradené vyčítaním uloženej hodnoty z FRAM pamäte. Hoci sa táto možnosť javí ako pomerne dobrý kompromis, problém nastáva v momente, ak príde u vypnutého systému k mechanickému pohybu osí vplyvom pôsobiacich vonkajších síl. V tom momente nie je hodnota uložená v FRAM ďalej platná, no riadiaci systém o tom nevie. Možnosť ukladania polohy do pamäte je preto vhodné využívať iba v aplikáciách, kde neprichádza ku manipulácii s vypnutou platformou.

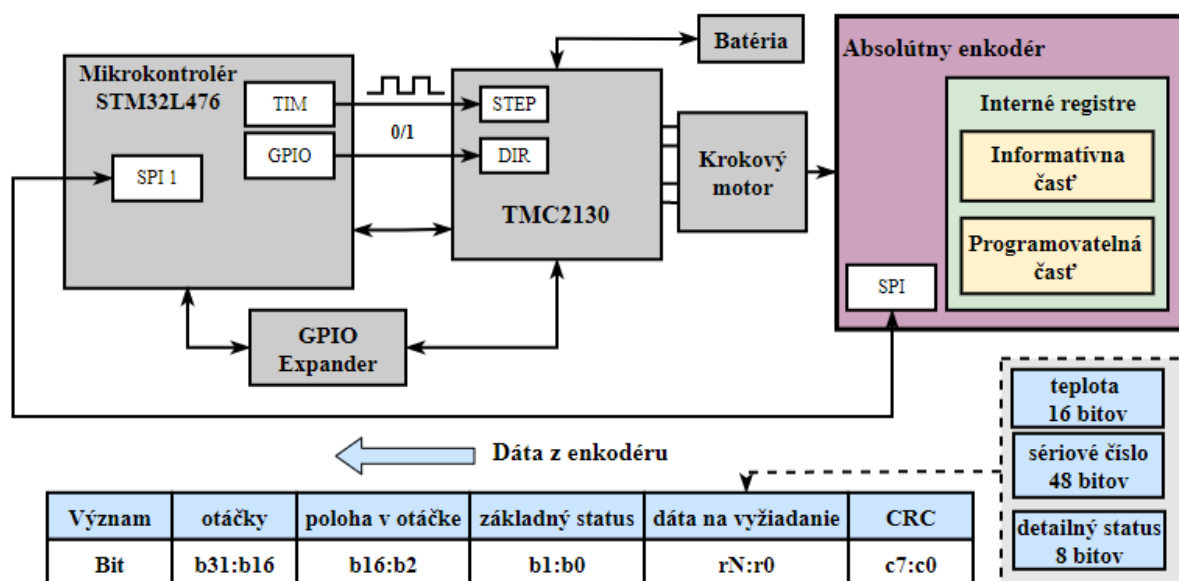
Jednou z funkcionalít radiča TMC2130 je tzv. StallGuard. Úlohou tejto patentovanej funkcie je detekovať strácanie krokov počas riadenia motorov. Vďaka tomu je možné po správnej kalibrácii použiť túto funkcionalitu ako náhradu koncových senzorov, keďže prináša možnosť detekcie stratených krokov pri mechanickom náraze na koniec osi. Z toho dôvodu je možné túto funkcionalitu využiť pri inicializačnom referencovaní osi.

Zvolený magnetický absolútny enkodér RLS Orbis komunikuje s mikrokontrolérom po zbernici SPI. Ako je uvedené v schéme na Obr. 7-15, interné registre tohto enkodéru je možné rozdeliť do dvoch častí. Registre informatívnej časti obsahujú dáta o absolútnom počte otáčok, polohe v rámci jednej otáčky a základný status, ktorého úlohou je informovať o platnosti dát a upozorniť na prípadné nebezpečné priblíženie sa limitom prevádzkových podmienok. Na vyžiadanie je možné získať doplnkové dáta ako sú teplota, špecifické sériové číslo definované výrobcom alebo detailný status. Úlohou detailného statusu je upozorniť na prekročenie limitu prevádzkových podmienok, ktorými sú príliš silné rušenie od externého magnetického poľa, prekročenie teplotného limitu, prípadne prekročenie maximálnej povolenej vzdialenosti budiaceho magnetu enkodéru od jeho Hallových čidiel.

Druhou časťou interných registrov je programovateľná časť. Medzi tento typ registrov patria offset polohy v rámci jednej otáčky, nastavenie čítača otáčok. Prostredníctvom špecifických programovacích sekvencií odoslaných cez SPI je možné tento typ interných registrov programovať na požadované hodnoty, vďaka čomu je možné definovať absolútnu nulovú polohu enkodéru.

Enkodér disponuje taktiež možnosťou automatickej kalibrácie, ktorej účelom je eliminovať vplyv prípadnej excentricity budiaceho magnetu voči Hallovým čidlám senzoru, ktorá môže vzniknúť nepresnou montážou.

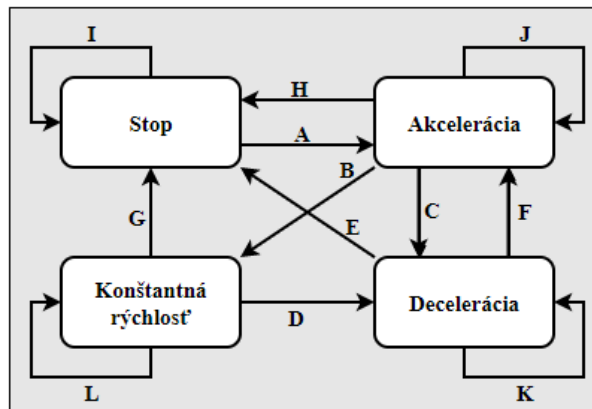
Výhodou je taktiež vlastná interná pamäť, ktorá slúži na uloženie jednotlivých nastavených parametrov, vďaka čomu nie je potrebné opakovat' nastavenia a kalibráciu.



Obr. 7-15 Schéma komunikácie enkodéru RLS ORBIS

7.7.3 Stavový automat krokového motora

Za účelom riadenia rýchlosti a polohy motora bol vytvorený stavový automat uvedený na Obr. 7-16. Jeho úlohou je generovanie trapézového, prípadne trojuholníkového rýchlostného profilu využitím definovaných limitných a požadovaných parametrov pohybu. V každom cykle tohto stavového automatu preto príde k výpočtu novej rýchlosti motora.



Obr. 7-16 Stavový automat na riadenie krokového motora

Zmena stavu môže nastať pri dosiahnutí niektorých z limitných, resp. požadovaných parametrov pohybu. Prechody G a H je možné označiť ako bezpečnostné zastavenia bez decelerácie. K týmto prechodom prichádza následkom signalizácie koncového senzora, pri dosiahnutí jednej z koncových polôh danej osi. Na zotrvanie v stave Stop, ktoré označuje vetva I, vplýva viacero podmienok riadiaceho algoritmu, ako sú napríklad dosiahnutie koncovkej polohy, blokovanie zariadenia (pripojená nabíjačka, pripojené USB...), prípadne dosiahnutie požadovanej polohy v móde absolútneho polohovania. Podmienky na zmenu, resp. zotrvanie v jednotlivých stavoch sú uvedené v Tab. 7-3.

Tab. 7-3 Podmienky zmien stavov automatu krokového motora

Zmena stavu	Dôvod zmeny stavu	
	Manuálne riadenie	Absolútne polohovanie
A	$PR > \check{S}R$	$(PP \neq AP) \ \& \ (PR > AR)$
B	$PR == AR$	$(PP \neq AP) \ \& \ (PR == AR)$
C	$PR < AR$	$(PP \neq AP) \ \& \ (ZD \leq DD)$
D	$PR < AR$	$(PP \neq AP) \ \& \ (ZD \leq DD)$
E	$AR == \check{S}R$	$(PP == AP)$
F	$PR > AR$	neexistuje
G	Jazdec na konci osi	Jazdec na konci osi
H	Jazdec na konci osi	Jazdec na konci osi
I	Definované v texte	Definované v texte
J	$PR > AR$	$(PP \neq AP) \ \& \ (ZD > DD) \ \& \ (PR > AR)$
K	$PR < AR$	$(PP \neq AP) \ \& \ (ZD \leq DD)$
L	$PR == AR$	$(PP \neq AP) \ \& \ (ZD > DD)$
Legenda	$\check{S}R$ – štartovacia rýchlosť PR – požadovaná rýchlosť AR – aktuálna rýchlosť	PP – požadovaná poloha AP – aktuálna poloha DD – dráha potrebná na deceleráciu ZD – zostávajúca dráha (rozdiel PP a AP)

Za účelom nastavenia novej vypočítanej rýchlosti je potrebné zmeniť frekvenciu časovaču slúžiaceho na generovanie STEP pulzov. Vzťah medzi frekvenciou generovaných pulzov a nastavením registrov príslušnej periférie, je daný rovnicou:

$$f_{STEP} = \frac{f_{clock}}{(PSC + 1) * (ARR + 1)} \quad (7-3)$$

Kde:

f_{STEP} – frekvenciu výstupného STEP signálu [1/s]

f_{clock} – taktovacia frekvencia mikrokontroléru [1/s]

PSC – Prescaler Register 16-bit (predelička) [-]

ARR – AutoReload Register 16-bit (perióda PWM cyklu) [-]

Z rovnice (7-3) vyplýva, že na frekvenciu časovača vplýva nastavenie hodnôt registrov predeličky (PSC) a AutoReloadRegistru (ARR). Keďže je potrebné určiť z jednej rovnice dve neznáme, ich nastavenie prebehne vo dvoch krokoch. V prvom kroku je uvažovaná maximálna možná hodnota ARR, nová hodnota registru PSC⁵ je následne určená ako:

$$PSC_{new} = \frac{f_{clock}}{f_{STEP} * (ARR_{max} + 1)} - 1 \quad (7-4)$$

Kde:

PSC_{new} – nová hodnota predeličky (Prescaler) [-]

ARR_{max} – maximálna hodnota AutoReload registru [-]

Po určení hodnoty PSC, je možné určiť hodnotu nastavenia ARR pomocou rovnice:

$$ARR_{new} = \frac{f_{clock}}{f_{STEP} * (PSC_{new} + 1)} - 1 \quad (7-5)$$

Kde:

ARR_{new} – nová hodnota AutoReload registru [-]

V prípade, že sú definované limity rýchlostí, je možné zafixovať hodnotu PSC na konštantnej hodnote, čo vedie ku zníženiu potrebných výpočtov.

Maximálna dosiahnuteľná frekvencia výstupného STEP signálu úzko súvisí s taktovacou frekvenciou mikrokontroléru a nastavením registrov PSC a ARR. Pri návrhu riadiaceho algoritmu je preto potrebné nájsť vhodný kompromis medzi hodnotou mikrokrokov, maximálnou požadovanou rýchlosťou pohonu a nastavenou taktovacou frekvenciou mikrokontroléru. V rámci tejto práce je využívaný mikrokontrolér s taktovacou frekvenciou 80 MHz a 16-bitovými registrami PSC a ARR.

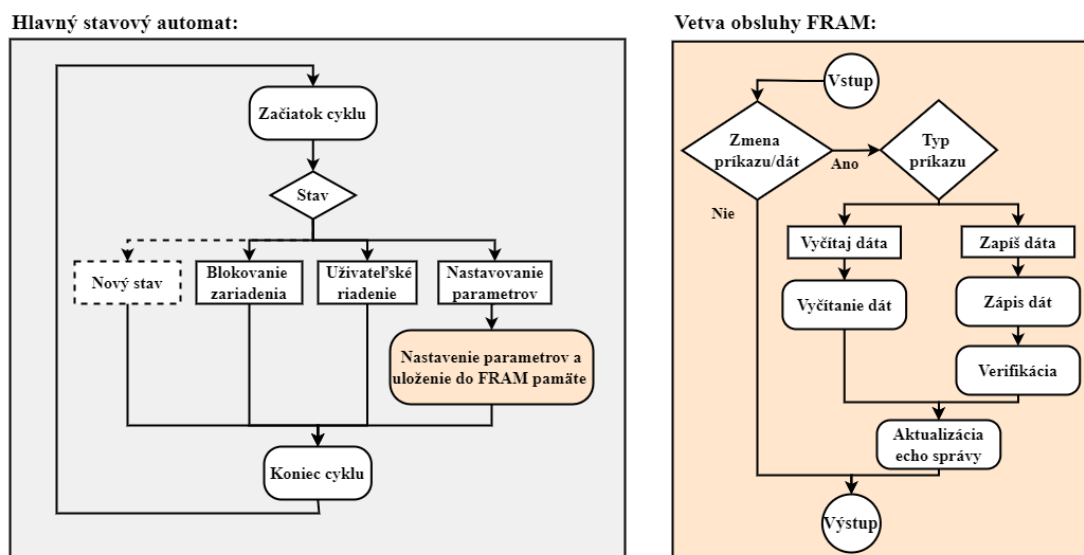
⁵ Delenie v rovniciach XX až YY je delenie celočíselných dátových typov. Z toho dôvodu prichádza ku zaokrúhľovaniu k najbližšiemu menšiemu výsledku.

7.8 Komunikácia s pamäťou FRAM

Pamäť FRAM je možné využiť na ukladanie rôznych parametrov a dát. V prípade riadiacej jednotky, pre ktorú je riadiaci algoritmus navrhnutý, ide predovšetkým o parametre rýchlostných profilov, ktoré môže užívateľ modifikovať. Možno je pri ďalšom vývoji taktiež vytvoriť funkcie na ukladanie rôznych diagnostických parametrov (napätie batérie, prijaté príkazy a podobne). V rámci návrhu riadiaceho algoritmu je potrebné vytvoriť spôsob jednoduchého zápisu, resp. vyčítania týchto dát po zbernici SPI, ktorá zabezpečuje prepojenie medzi FRAM a mikrokontrolérom.

7.8.1 Obsluha pamäte FRAM

Ku samotnej obsluhu FRAM prichádza v jednej z vetví hlavného stavového automatu. Ako uvádza schéma na Obr. 7-17, pri zápise dát je potrebná verifikácia ich správneho zápisu. Za týmto účelom sú zapísané dáta opäť vyčítané a porovnané s pôvodnou zapisovanou hodnotou. Ak príde k úspešnému zápisu, prichádza ku odoslaniu nastavených dát prostredníctvom echo správy. Vďaka tomu užívateľ získava informáciu o úspešnom nastavení požadovaných dát.



Obr. 7-17 Diagram obsluhy pamäte FRAM

7.8.2 Princíp adresovania

Každému z dát uložených v FRAM je definovaná začiatková a koncová adresa, ktorá závisí od ich dĺžky. Jednotlivé adresy sú definované v knižnici na obsluhu FRAM ako enumerátor. Vďaka tomuto prístupu je zabezpečené prehľadnejšie adresovanie jednotlivých premenných a lepšia adaptácia na prípadné zmeny. Taktiež sú v rámci knižníc definované funkcie na zápis, resp. vyčítanie konkrétnych dát, ktoré tieto využívajú tieto adresy. Vďaka univerzálnemu prístupu ku práci s FRAM je možné pri ďalšom vývoji relatívne jednoduché rozširovanie o funkcie slúžiace na ukladanie nových parametrov.

7.9 Integrácia doplnkových komponentov

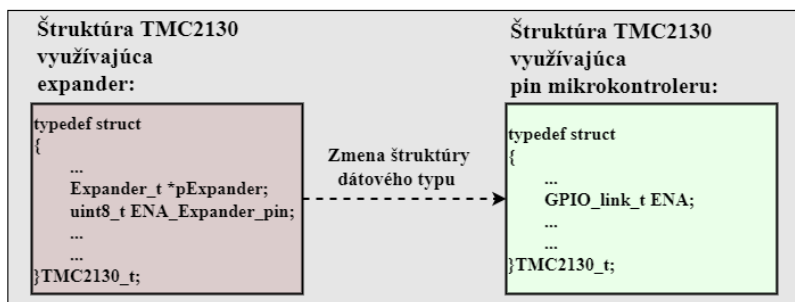
Nasledujúce podkapitoly popisujú spôsob integrácie riadenia komponentov do riadiaceho algoritmu, ktoré vybral v rámci svojej práce Ondryáš [39].

7.9.1 Riadenie GPIO expandéru

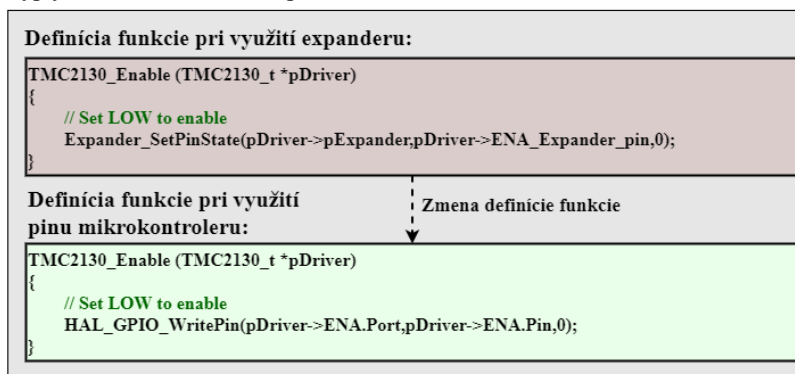
Ako uvádza Ondryáš [39], myšlienka využitia GPIO expandéru vznikla z dôvodu nedostatku mikrokontrolérov na trhu. Hlavnou úlohou expandéru je ovládanie, resp. detekcia stavu digitálnych pinov jednotlivých komponentov, ktoré sú k nemu pripojené. Vďaka tomu je možné v prípade nedostupnosti aktuálne využívaného mikrokontroléru použiť alternatívny mikrokontrolér disponujúci menším počtom digitálnych pinov, čo je možné považovať za hlavnú výhodu využitia expandéru. Medzi čiastočnú nevýhodu patrí nutnosť komunikácie prostredníctvom SPI s mikrokontrolérom pri riadení stavu pinov. Z toho dôvodu nie je expandér vhodný na riadenie digitálnych pinov, ktoré vyžadujú rýchle a časté spínanie.

Za účelom udržania požiadavky na univerzálnosť a robustnosť riadiaceho algoritmu bol pri definícii riadiacich funkcií digitálnych pinov jednotlivých komponentov zvolený nasledujúci prístup. Ak je komponent riadiacej jednotky prepojený s GPIO expandérom, je potrebné prepojenie ich vytvorených dátových štruktúr, ako uvádza kapitola 7.2.1. Obr. 7-18 zobrazuje príklad riadenia digitálneho pinu ENA radiča TMC2130, ktorý je v riadiacej jednotke pripojený ku GPIO expandéru. V prípade odstránenia expandéru z riadiacej jednotky by bola potrebná jednoduchá zmena dátovej štruktúry na obsluhu radiča a zmena definície funkcie, tak ako je uvedené na Obr. 7-18. Digitálny pin by mohol byť následne obsluhovaný priamo GPIO perifériou mikrokontroléru. Navrhnuté riešenie zvyšuje robustnosť algoritmu v prípade hardwarových zmien riadiacej jednotky, či už ide o zmenu využívaných pinov expandéru, alebo jeho úplné odstránenie.

Vplyv odstránenia GPIO expandéru na zmenu dátovej štruktúry:



Vplyv odstránenia GPIO expandéru na zmenu definície funkcie:



Obr. 7-18 Porovnanie využitia GPIO expandéru a digitálneho pinu mikrokontroléru

7.9.2 IMU senzor

Polohovacie platformy, pre ktoré je tento bezdrôtový systém navrhovaný, je možné využívať v zariadeniach kde prichádza ku rotácii platformy okolo vertikálnej osy. Ako uvádza Ondryáš [39], jednou z navrhovaných funkcionalít systému je informovanie užívateľa o aktuálnom natočení platformy v zariadení. Z toho dôvodu bol zvolený IMU senzor ICM-20789, ktorý disponuje trojosím gyroskopom a akcelerometrom. Komunikácia s mikrokontrolérom je zabezpečená po zbernici I2C.

S cieľom získať hodnoty natočenia je potrebné získané hodnoty uhlovej rýchlosti integrovať. Pri integrovaní uhlovej rýchlosti sa obecné môžeme stretnúť s problémom driftovania hodnôt. Za účelom kompenzácie týchto chýb je možné využívať komplementárne filtre, ktoré pracujú na fúzií dát z dvoch, alebo viacerých senzorov. Najčastejšie sa môžeme stretnúť s fúziou meraných dát akcelerometrom a gyroskopom. Tento prístup však nie je využiteľný pri rotácii v horizontálnej rovine, keďže vektor tiažového zrýchlenia zostáva paralelný s osou rotácie.

Aktuálne navrhnuté riešenie aplikuje základnú aktiváciu interného low-pass filtru senzora podľa dokumentácie výrobcu, a jednoduchú numerickú integráciu meraného uhlového zrýchlenia v osi rotácie. Získanú hodnotu natočenia je možné z riadiacej jednotky vyčítavať pomocou príkazu a vykresľovať v užívateľskom software. Komplexné riešenie problematiky odmeriavania rotácie platformy bude už predmetom ďalšieho vývoja.

7.9.3 Výkonový modul

Výkonový modul disponuje viacerými funkcionalitami. Medzi základné funkcionality patrí napájanie riadiacej jednotky a pohonov. Taktiež v sebe integruje nabíjací obvod batérie. Medzi ďalšie funkcionality patrí odmeriavanie napätia batérie a veľkosti nabíjacieho prúdu, ktorých hodnoty sú získané využitím ADC prevodníkov mikrokontroléru.

Ako uvádza Ondryáš [39], pre výkonový modul pri návrhu riadiacej jednotky vytvoril taktiež rozšírené funkcionality, ako sú možnosť firmwarového odpojenia nabíjacieho obvodu, detekcia pripojenej nabíjačky, možnosť firmwarového vypnutia systému pomocou tzv. „KillSwitch“ pinu. Pôvodný plán bolo taktiež vytvorenie možnosti „Sleep“ módu, kedy by prišlo k odpojeniu vybraných periférií softwarovým príkazom za účelom šetrenia batérie v prípade nečinnosti platformy. Keďže prišlo počas návrhu riadiacej jednotky ku chybné voľbe tranzistorov, táto funkcionalita bude dostupná až na novej riadiacej jednotke, ktorá je v čase písania tejto práce už vo výrobe.

Merané hodnoty napätia batérie a nabíjacieho prúdu je možné vyčítavať definovanými príkazmi z užívateľského softwaru.

7.9.4 Odmeriavanie teploty

Prekročenie prevádzkových limitov teploty môže mať za následok nesprávnu funkciu komponentov riadiacej jednotky, alebo viesť až ku ich poruchám. Odmeriavanie teploty preto zohráva dôležitú úlohu z hľadiska diagnostiky správnej funkčnosti jednotlivých komponentov, ktoré sa počas svojej prevádzky zahrievajú. Ide napríklad o nabíjací modul, batériu, alebo radiče krokových motorov. Zvýšená teplota ovplyvňuje taktiež kvalitu meraných dát z enkodérov alebo IMU jednotky.

Väčšina komponentov, ktoré riadiaca jednotka využíva obsahuje interné odmeriavanie teploty. Na prekročenie teplotných limitov reagujú tieto komponenty prechodom do tzv. „*Thermal Shutdown*“ módu, prípadne odoslaním informácie o prekročení limitu do mikrokontroléru. Dosahovanie teplotných limitov komponentov počas prevádzky však nie je žiadúce, keďže vedie ku skresľovaniu meraných dát, skracovaniu životnosti, prípadne poruche riadiacej jednotky. Z toho dôvodu sa Ondryáš [39] rozhodol zakomponovať do návrhu riadiacej jednotky senzor teploty TMP117, ktorý s mikrokontrolérom komunikuje prostredníctvom rozhrania I2C.

Úlohou riadiaceho algoritmu je vyčítavanie a spracovanie dát tohto teplotného senzoru a odosielanie ich hodnoty do užívateľského softwaru. Predmetom ďalšieho vývoja je interná kontrola teploty a automatická reakcie na presiahnutie nastavených teplotných limitov.

7.9.5 Užívateľské tlačidlo s RGB LED

Súčasťou riadiacej jednotky je užívateľské tlačidlo doplnené RGB LED krúžkom. Tlačidlo slúži na zapnutie/vypnutie zariadenia, prípadne resetovanie. Výhodou tlačidla je jeho vlastný kontrolér. Vďaka tomu je po užívateľskom príkaze na vypnutie možné vykonať vypínaciu sekvenciu, do ktorej môže patriť uloženie požadovaných dát do pamäte FRAM. Vypnutie zariadenia je zabezpečené mikrokontrolérom pomocou „KillSwitch“ pinu ako je uvedené v 7.9.3. RGB LED krúžok slúži na hardwarovú signalizáciu stavu zariadenia. Súvislosť medzi stavmi zariadenia a stavmi LED je uvedená v Tab. 7-4.

Tab. 7-4 Signalizačné sekvencie RGB LED krúžku

Signalizácia stavu	Farba LED	Stav LED
Pripojená nabíjačka	Červená	Zapnutá LED
Slabá batéria	Červená	Rýchle blikanie LED
„Stand-by“ stav po zapnutí	Zelená	Zapnutá LED
Jednotka pripravená na spárovanie s PC	Modrá	Pomalé blikanie LED
Riadiaca jednotka komunikuje s PC	Modrá	Zapnutá LED

8 Návrh užívateľského PC softwaru

Navrhovaný PC software má za úlohu poskytnúť užívateľovi jednoduché a prehľadné grafické prostredie využívané pri riadení polohovacej platformy. Software je napísaný v jazyku C#. Na tvorbu grafického rozhrania využíva knižnicu Windows Forms, ktorá je súčasťou .NET frameworku. V nasledujúcich podkapitolách sú postupne popísané požiadavky na software, jednotlivé bloky navrhnutého užívateľského prostredia a základný postup pri tvorbe príkazových, resp. spracovávaní echo správ.

8.1 Požiadavky na software

Podobne ako u návrhu riadiaceho algoritmu, je pri návrhu užívateľského PC softwaru potrebné zohľadniť požiadavky na funkcionality systému, ktoré stanovil vo svojej práci Ondryáš [39].

Pri návrhu softwaru je preto potrebné zohľadniť nasledujúce požiadavky:

- Manuálne riadenie pohonov – kontinuálny pohyb zvolenou rýchlosťou
- Automatické, absolútne a relatívne⁶ polohovanie
- Nastavovanie parametrov a dynamiky polohovania
- Automatické vyhľadanie a pripojenie riadiacej jednotky (BLE komunikácia)
- Tvorba príkazových správ a spracovanie echo správ riadiacej jednotky
- Informácie o stave zariadenia (status, stav batérie, aktuálna poloha...)
- Jednoduché rozširovanie o ďalšie funkcionality
- Vytvorenie vývojárskych funkcionalít zjednodušujúcich ďalší vývoj systému

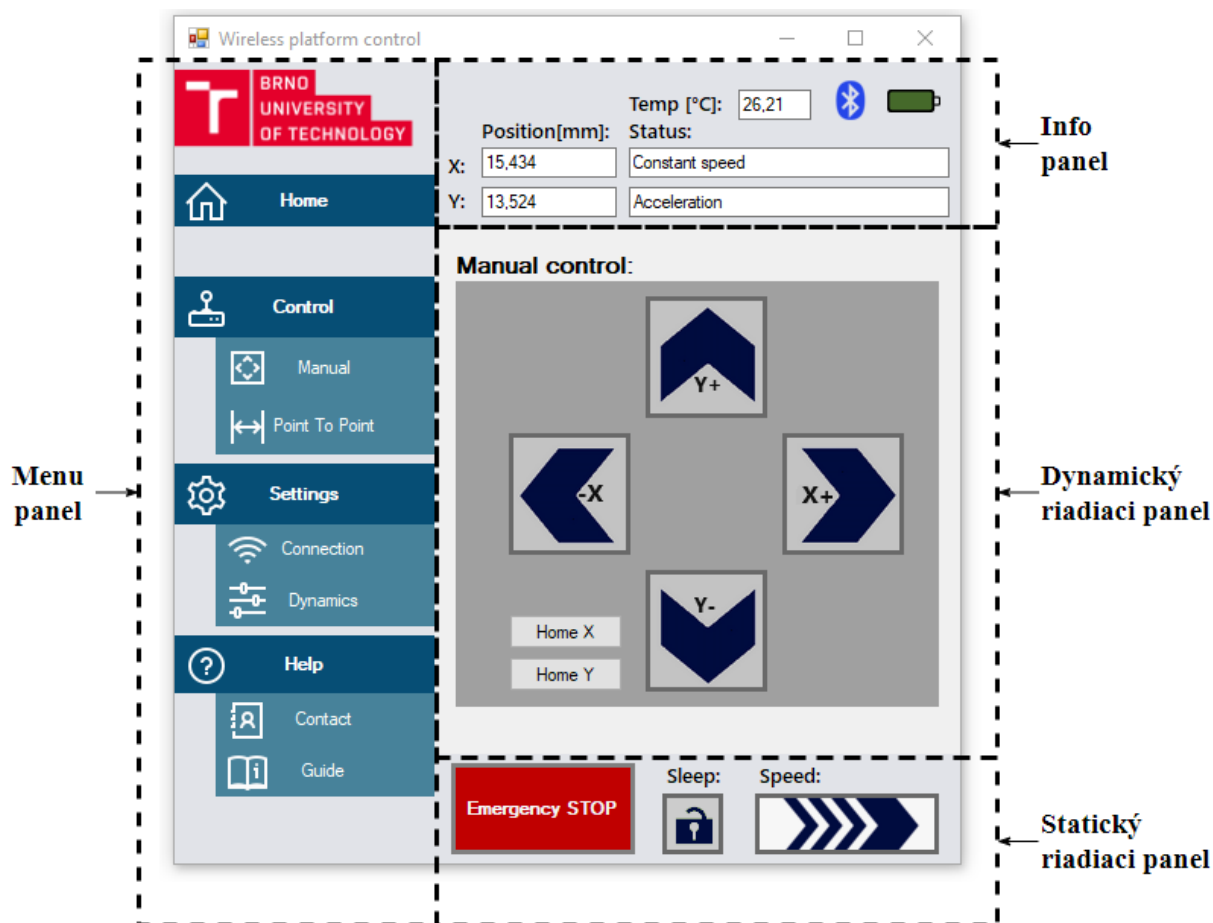
⁶ Relatívne polohovanie sa často označuje ako Jog mód. Prichádza v ňom ku pohybu v definovaných inkrementoch.

8.2 Rozloženie užívateľského prostredia

Základné užívateľské prostredie je možné rozdeliť do štyroch hlavných častí uvedených na Obr. 8-1. Info panel slúži na zobrazovanie najpodstatnejších informácií. Pre obe osi je v ňom zobrazovaná aktuálna pozícia a taktiež status, ktorý je možné chápať ako informáciu o aktuálnom stave osi. Pohon osi môže byť v pohybe (akcelerácia/decelerácia/konštantná rýchlosť), dosiahnuť požadovanú polohu, byť manuálne zastavený, prípadne blokován z dôvodu nízkej batérie alebo pripojenej nabíjačky. O všetkých spomenutých možnostiach stavu osi je užívateľ informovaný práve prostredníctvom statusu. Info panel ďalej obsahuje aktuálnu teplotu riadiacej jednotky, signalizáciu aktívneho Bluetooth pripojenia a taktiež signalizáciu stavu batérie.

Menu panel slúži na zmenu dynamického riadiaceho panelu. Ten môže podľa voľby užívateľa obsahovať možnosť manuálneho riadenia, automatického polohovania do absolútnej, resp. relatívnej polohy a taktiež nastavovanie dynamiky osí. Bližšie je dynamický riadiaci panel popísaný v podkapitole 8.2.1.

Poslednou časťou základného užívateľského prostredia je statický riadiaci panel. Ako vyplýva z jeho názvu, užívateľ jeho obsah nemení, no môže využívať príslušné tlačidlá. Popis jeho jednotlivých častí je uvedený v podkapitole 8.2.2.



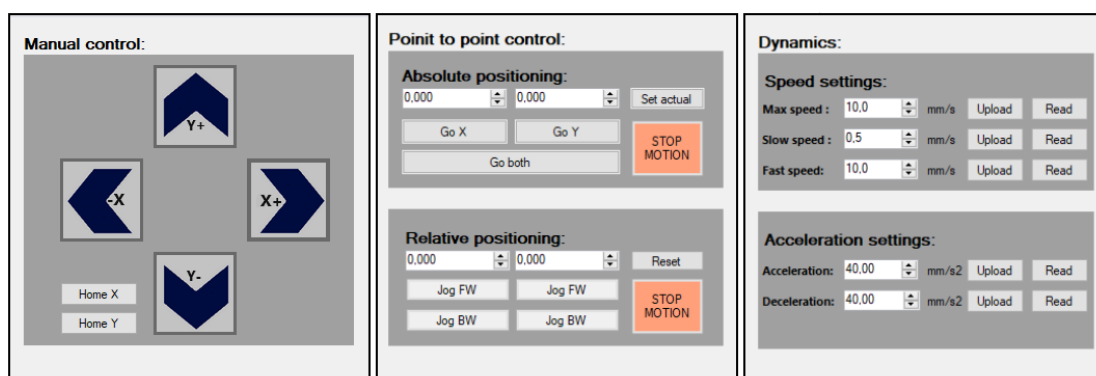
Obr. 8-1 Rozloženie základného užívateľského prostredia

8.2.1 Dynamický riadiaci panel

Jednou z požiadaviek kladených na návrh softwaru je udržať jednoduché užívateľské prostredie. Z toho dôvodu bolo potrebné navrhnuť intuitívne a prehľadné riešenie. Podľa potreby si užívateľ môže zvoliť medzi manuálnym a automatickým polohovaním. Taktiež je možné nastavovať dynamické parametre zariadenia. Vizualizácie jednotlivých častí softwaru sú uvedené na Obr. 8-2.

U manuálneho polohovania užívateľ zadáva príkazy pomocou myši, prípadne pomocou smerových kláves na klávesnici, čo prináša možnosť ovládania platformy v oboch smeroch súčasne. V prípade potreby je možné odoslať príkaz na kalibračné referencovanie osi kliknutím na tlačidlo Home X, resp. Home Y. Automatické polohovanie je rozdelené na absolútne polohovanie a relatívne polohovanie. Zatiaľ čo u absolútneho polohovania je potrebné definovať absolútnu cieľovú polohu, relatívne polohovanie je založené na pričítavaní užívateľom definovaného inkrementu ku aktuálnej polohe platformy.

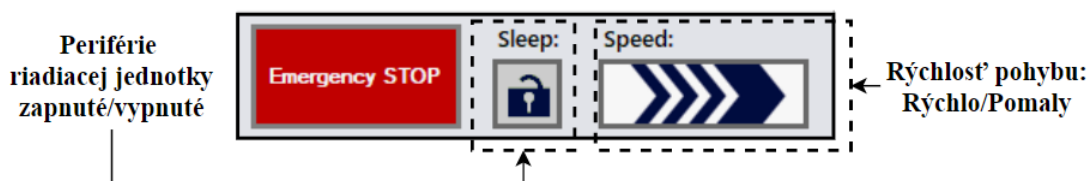
V časti nastavenia dynamiky je možné nastavovať hodnoty rýchlej a pomalej rýchlosti, ktoré sú bližšie popísané v podkapitole 8.2.2. Taktiež je možné nastavovať akceleráciu pohybu. Všetky nastavené veličiny sú automaticky ukladané do pamäte FRAM riadiacej jednotky, preto nie je potrebné opakovať ich nastavenie po opakovanom spustení softwaru.



Obr. 8-2 Možnosti dynamického riadiaceho panelu

8.2.2 Statický riadiaci panel

Súčasťou statického riadiaceho panelu na Obr. 8-3 je nastavenie rýchlej a pomalej rýchlosti, ktorých konkrétne parametre môže užívateľ definovať v nastavení dynamiky ako je uvedené v kapitole 8.2.1. Kliknutím na tlačidlo „Sleep“ prichádza ku odoslaniu príkazovej správy na vypnutie vybraných periférií riadiacej jednotky, čím je poskytnutá možnosť šetrenia batérie. Ako uvádza podkapitola 7.9.3, plne funkčný „Sleep mode“ bude dostupný až v novej verzii riadiacej jednotky. Počas tohto módu prichádza taktiež k zablokovaniu ovládacích tlačidiel softwaru.



Obr. 8-3 Statický riadiaci panel

8.3 Vývojárske rozšírenie

Užívateľské prostredie je možné rozšíriť o vývojársku časť jednoduchou klávesovou skratkou *Shift+V*. Veľkou výhodou tohto rozšírenia je možnosť simulácie rôznych typov prijímanej echo správy. Taktiež je možné spustiť simulačný časovač, ktorý cyklicky číta príkazy z užívateľského prostredia a simuluje odosielanú správu. Vďaka tomuto riešeniu je možné testovať jednotlivé funkcionality softwaru bez potreby pripájania riadiacej jednotky, čo výrazne uľahčuje vývoj. Taktiež je možné otestovať správnosť softwarových funkcií pred pripojením k riadiacej jednotke, čím sa predchádza možnému poškodeniu systému v prípade nesprávneho riadenia.

Simulované echo na Obr. 8-4 obsahuje nastavenie statusu oboch osí indikujúceho pripojenie nabíjačky a aktuálnu polohu. V info paneli je možné vidieť správnu odozvu na simulovanú echo správu.

Odozva na simulovanú echo správu

Temp [°C]: 25.36

Position[mm]:
X: 75.258
Y: 12.457

Status:
Blocked - Charger Connected
Blocked - Charger Connected

Manual control:

Y+

-X

X+

Y-

Home X

Home Y

Emergency STOP

Sleep:

Speed:

Vývojárske rozšírenie

Real-Time Plot

Developer_area

Command
Type: TMC2130_SET_IRUN_CURRENT
Value: 0
TMC2130_SET_IRUN_CURRENT
Value: 0
Subscribers
Type: SubsID_Battery_value
Value: SubsID_Battery_value
SetSubscriber
PrintCommand

Echo simulation
Type: ECHO_MOTOR_POSITION_STATUS_BLOCKED_CHARGER_CO
Value: 75258
ECHO_MOTOR_POSITION_STATUS_BLOCKED_CHARGER_CO
Value: 12457
SubsID_Battery_value
Value: 15236
SubsID_Temperature
Value: 2536
SetEcho
0029|075258|0029|012457|0005|015236|0009|002536
Start Test Stop Test

Simulácia echo správy

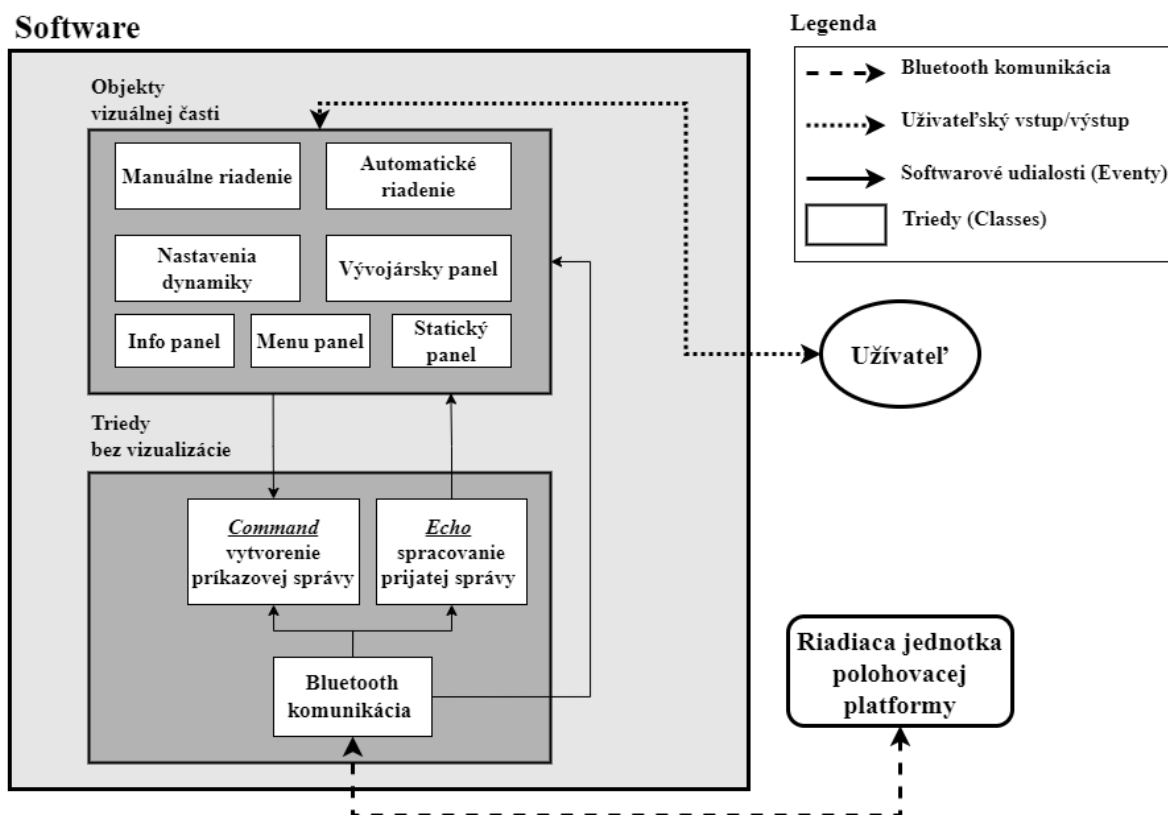
Obr. 8-4 Vizualizácia vývojárskeho rozšírenia

8.4 Triedy užívateľského softwaru

Jazyk C# patrí medzi objektovo orientované programovanie jazyky. Z toho dôvodu je potrebné navrhnuť triedy, ktoré slúžia na inicializáciu jednotlivých využívaných objektov. Ich súčasťou sú premenné, metódy a udalosti. S cieľom zabezpečiť modularitu riešenia sú pre jednotlivé časti softwaru vytvorené triedy, ktoré sú medzi sebou prepojené práve pomocou metód a udalostí. V rámci tejto práce je možné ich rozdelenie do dvoch častí zobrazených na Obr. 8-5.

Triedy prvej časti je možné označiť ako vizuálne, keďže sú využívané na vytváranie objektov zabezpečujúcich priamu interakciu medzi užívateľom a softwarom. Objekty, ktoré sú pomocou týchto tried inicializované obsahujú tlačidlá, textové polia, a rôzne iné prvky. V rámci jednotlivých tried sú taktiež definované metódy na spracovanie príslušných užívateľských vstupov, resp. nastavenie výstupov. Príklady vytvorených vizuálnych objektov je možné vidieť v predchádzajúcich podkapitolách.

Do druhej časti patria triedy, ktoré slúžia na zabezpečenie komunikácie medzi softwarom a riadiacou jednotkou. Neobsahujú preto žiadne vizuálne prvky a ich jednotlivé úlohy bežia na pozadí aplikácie. Hlavným účelom týchto tried je tvorba príkazových správ, zabezpečenie ich odosielania prostredníctvom Bluetooth a po prijatí echo správy túto správu spracovať a ďalej odoslať vizuálnym objektom. Obsahujú taktiež definíciu jednotlivých identifikátorov správ, ktoré boli popísané v kapitole 6.



Obr. 8-5 Rozdelenie vytvorených softwarových tried

8.5 Bluetooth komunikácia

Pri návrhu triedy na riadenie Bluetooth komunikácie je potrebné zohľadniť typ komunikačného modulu, ktorý využíva riadiaca jednotka. Vývojová riadiaca jednotka uvedená v podkapitole 5.1 využívala hobby modul HC-05. Keďže tento modul používa klasické Bluetooth rozhranie, je možné ho s užívateľským PC spárovať relatívne jednoducho. Komunikácia následne prebieha po sériovej linke prostredníctvom virtuálneho sériového portu.

Prototypové riešenia riadiacej jednotky uvedené v podkapitole 5.2 však využíva na komunikáciu BLE modul, ktorého obsluha je pomerne komplikovanejšia v porovnaní s klasickým Bluetooth. Z toho dôvodu je potrebné vytvoriť softwarové riešenie, ktoré je schopné tento modul nájsť, zabezpečiť jeho pripojenie a riadiť tok správ.

Jedným z cieľov práce Ondryáša [39] bolo overenie funkčnosti riadiacej jednotky. Z toho dôvodu v rámci svojej práce vytvoril triedu, v rámci ktorej definoval metódy pre overenie základných funkcionalít použitého BLE modulu (spárovanie, odoslanie a prijatie správy...).

V rámci tejto práce, bola táto trieda rozšírená o udalosti slúžiace na prepojenie s ostatnými objektami užívateľského softwaru. Taktiež bolo potrebné jej prispôbiť požiadavkám navrhnutého komunikačného protokolu uvedeného v kapitole 6. Keďže táto trieda obsahuje viacero komplikovaných metód, Tab. 8-1 uvádza iba zoznam základných metód, ktoré sú dôležité pre jej ďalšie využitie v navrhovanom software.

Tab. 8-1 Základné metódy triedy BLE_comm

Názov metódy	Význam
BLE_StartScanning	Spustiť vyhľadávanie BLE modulu
BLE_StopScanning	Zastaviť vyhľadávanie BLE modulu
BLE_ConnectDevice	Odoslanie požiadavku na prepojenie BLE modulu s PC
BLE_DisconnectDevice	Odoslanie požiadavku na odpojenie BLE modulu od PC
BLE_SendMessage	Príkaz na odoslanie správy z PC do BLE modulu
BLE_ReceivedMessage	Poslucháč prijímaných správ do PC.

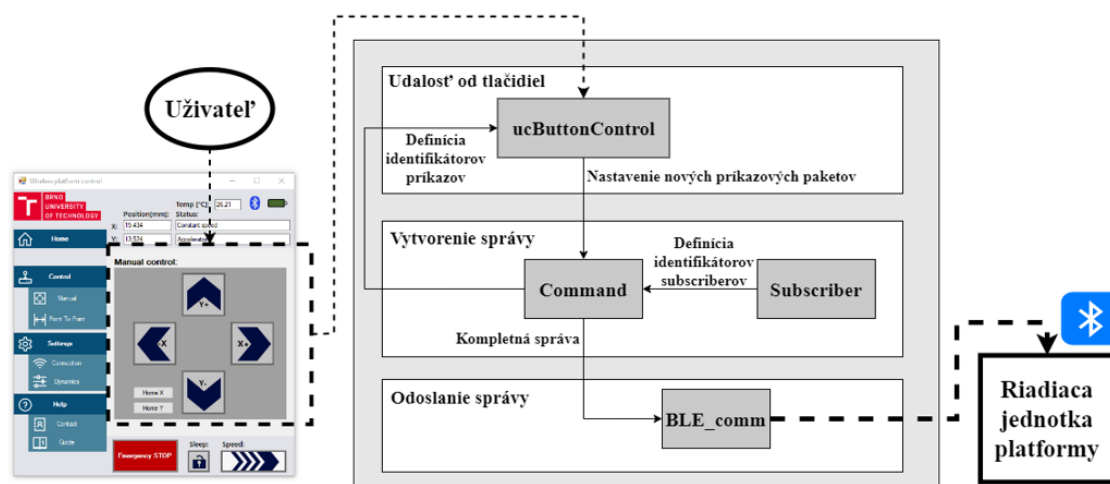
Udalosti uvedené v Tab. 8-2 je možné prepojiť s metódami ďalších objektov riadiaceho softwaru.

Tab. 8-2 Udalosti definované v triede BLE_comm

Názov udalosti	Význam
BleDeviceStartScanningEvent	Začiatok vyhľadávania zariadenia
BleDeviceIdFoundEvent	Zariadenie je v dosahu PC
BleDeviceConnectedEvent	Zariadenie je úspešne pripojené k PC
BleDeviceDisconnectingEvent	Zariadenie sa odpája
BleDeviceDisconnectedEvent	Zariadenie je odpojené

8.6 Proces odoslania príkazovej správy

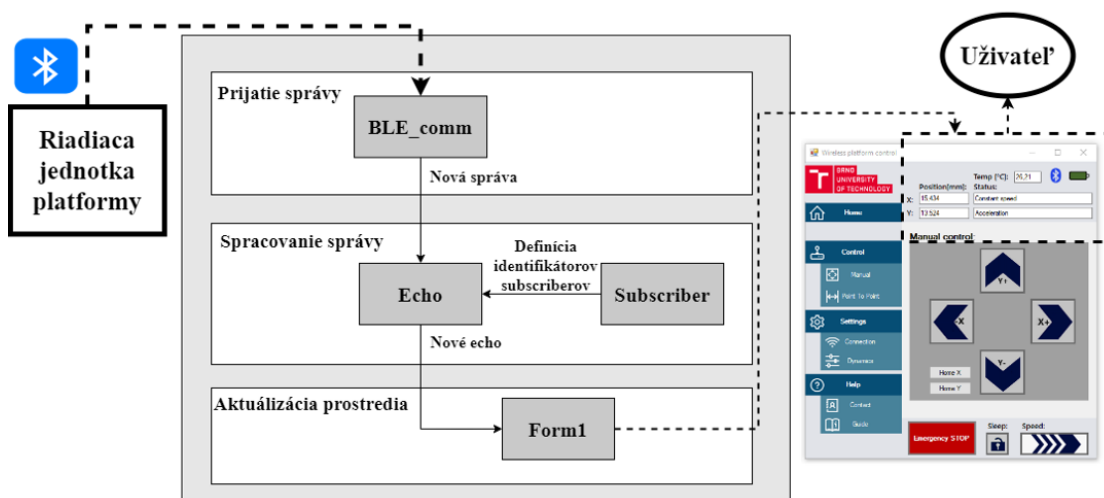
Schéma procesu odoslania príkazovej správy na Obr. 8-6 uvádza príklad manuálneho riadenia. Jednotlivé príkazové identifikátory sú definované v rámci triedy *Command*. Užívateľské vstupy (udalosti od tlačidiel) sú obslužené metódami definovanými v rámci triedy *ucButtonControl*. Úlohou týchto metód je nastaviť nové príkazové pakety obsahujúce konkrétne príkazové identifikátory a príslušné dáta. Účelom triedy *Subscriber* je definovať jednotlivé identifikátory subscriberov, ktorých význam bol popísaný v podkapitole 6.3. Kompletná správa skladajúca sa z príkazových paketov a identifikátorov subscriberov slúži ako vstupný argument pre metódu obsluhujúcu samotnú Bluetooth komunikáciu, ktorá je definovaná v rámci triedy *BLE_comm*.



Obr. 8-6 Schéma procesu odoslania príkazovej správy

8.7 Proces vyhodnocovania echo správy

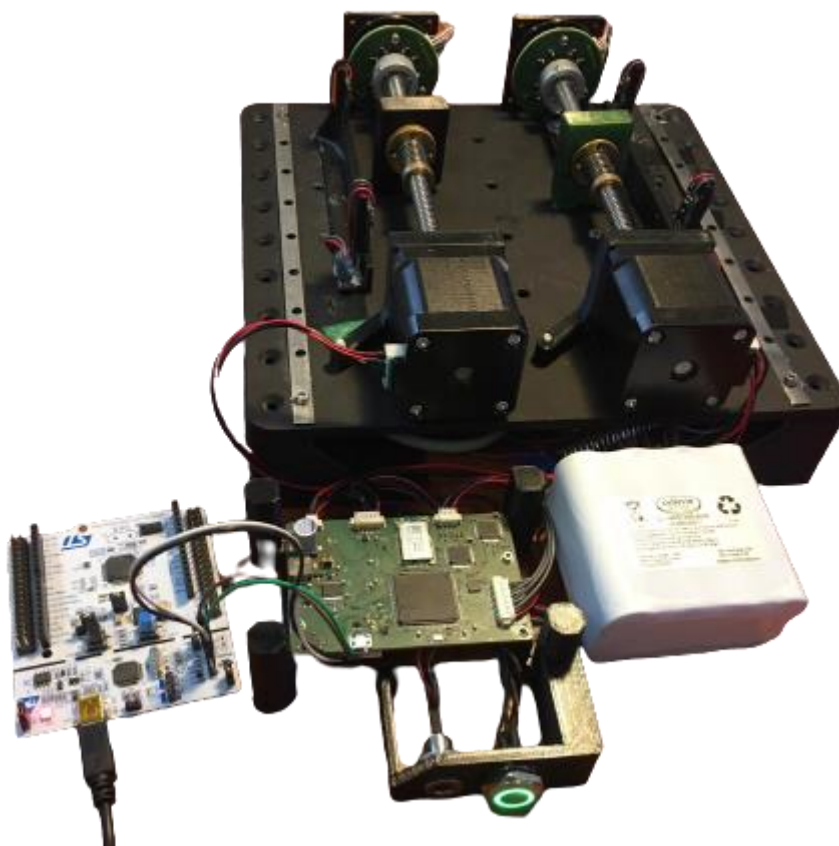
Riadiaca jednotka platformy odosiela do užívateľského PC echo správy. Samotné prijatie správy je zabezpečené metódami definovanými v rámci triedy *BLE_comm*. Po prijatí reťazca s požadovaným počtom znakov prichádza ku jeho spracovaniu metódami definovanými v rámci triedy *Echo*. Výsledkom spracovania je vytvorenie nového echo objektu, ktorý je argumentom udalosti *NewEchoEvent*. V rámci triedy *Form1*, sú definované metódy na spracovanie nového echo objektu slúžiace na aktualizáciu užívateľského prostredia.



Obr. 8-7 Schéma procesu spracovania echo správy

9 Testovanie navrhnutého riadiaceho algoritmu

Na overenie funkcionality riadiaceho algoritmu bola v rámci tejto práce vytvorená testovacia zostava uvedená na Obr. 9-1. Táto zostava neslúži na polohovanie, ale iba na verifikáciu funkcionality navrhovaného systému. Z toho dôvodu bolo možné uložiť motory oboch osí paralelne za účelom udržania kompaktnosti testovacieho systému. Na zhotovenie sú využité komponenty uvedené v podkapitole 4.3. Testovacia zostava je pripojená ku riadiacej jednotke napájanej batériou, ktorú navrhol Ondryáš [39]. Za účelom priamej akvizície dát z mikrokontroléru je riadiaca jednotka prepojená s ST-Link debuggerom, ktorý je súčasťou vývojovej dosky Nucleo uvedenej v podkapitole 5.1. Vďaka tomuto prepojeniu je možné v softvare STM Studio ukladať hodnoty zvolených premenných, ktoré vystupujú v riadiacom algoritme. Na vykreslenie priebehu sledovaných veličín je využitý MATLAB. Priebeh digitálnych signálov je získaný pomocou logického analyzátora prepojeného so softwarom Logic 2 od spoločnosti Saleae. V nasledujúcich podkapitolách sú uvedené výsledky testovania riadenia pohonov, komunikácie s pamäťou FRAM, zápis do registrov radiča TMC2130 a užívateľského softwaru.



Obr. 9-1 Testovacia zostava s riadiacou jednotkou

9.1 Testovanie riadenia krokových motorov

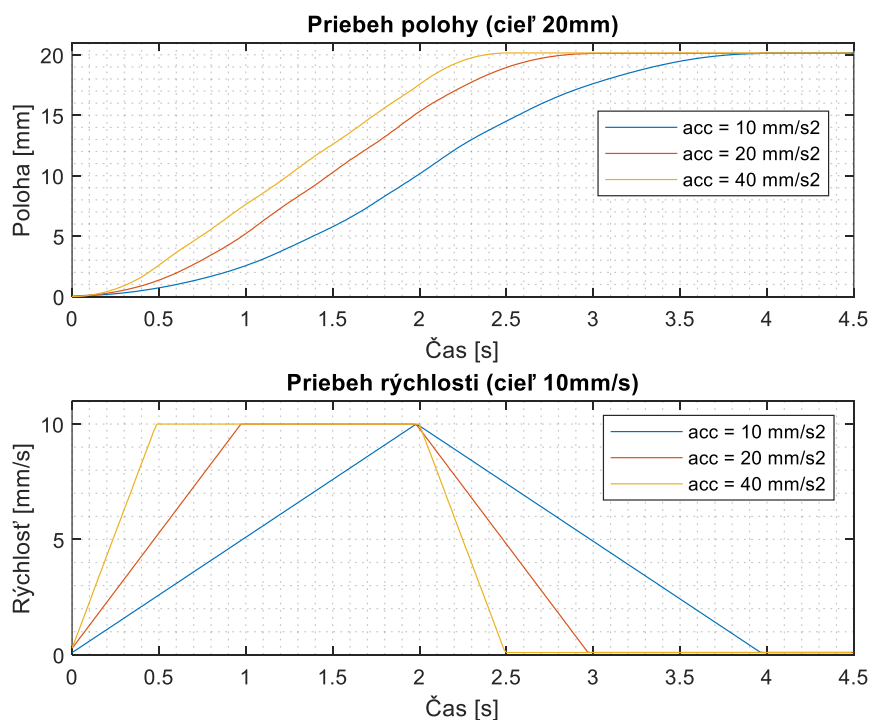
Ako je spomenuté v kapitole 7.7.2, krokové motory sú riadené bez spätnej väzby. Z toho dôvodu je možné zvolený enkodér, ktorý je súčasťou testovacej zostavy na Obr. 9-1 využiť na verifikáciu automatického polohovania s definovanou cieľovou polohou, ktorá je uvedená v podkapitole 9.1.1.

V móde manuálneho polohovania môže užívateľ meniť požadovaný smer pohybu počas chodu motoru. Keďže je smer otáčania motora riadený digitálnym signálom, je možné zmeniť smer otáčania prakticky okamžite. Toto správanie sa však nie je žiadúce, keďže by prichádzalo k mechanickým rázom, znižovaniu životnosti a prípadnej strate krokov. Z toho dôvodu bolo potrebné navrhnuť systém manuálneho riadenia tak, aby v prípade požiadavky na zmenu smeru počas chodu motora, prišlo najskôr k decelerácii a až následne ku zmene smeru otáčania a akcelerácii na požadovanú rýchlosť. Výsledky testovania tohto riadenia sú uvedené v podkapitole 9.1.2

9.1.1 Testovanie akcelerácie pri automatickom polohovaní

Jednou z požiadaviek na riadiaci systém je možnosť užívateľského nastavovania akcelerácie pohonu. Táto funkcionality bola testovaná v móde absolútneho polohovania, kedy bola zo softwaru odoslaná požiadavka absolútneho polohovania do vzdialenosti 20 mm, ktorej predchádzalo nastavenie požadovanej akcelerácie. Uvedené priebehy polohy boli získané meraním pomocou pripojeného enkodéru. Vďaka tomu bola overená správna funkčnosť riadenia bez spätnej väzby.

Na druhom grafe je uvedený priebeh rýchlostí, ktoré boli získané prepočítavaním nastavenej frekvencie časovaču na generovanie STEP pulzov. Na uvedenom grafe je možné vidieť saturáciu na požadovanej hodnote rýchlosti 10mm/s. Taktiež je možné sledovať vplyv zmeny akcelerácie nastavovanej z užívateľského softwaru.

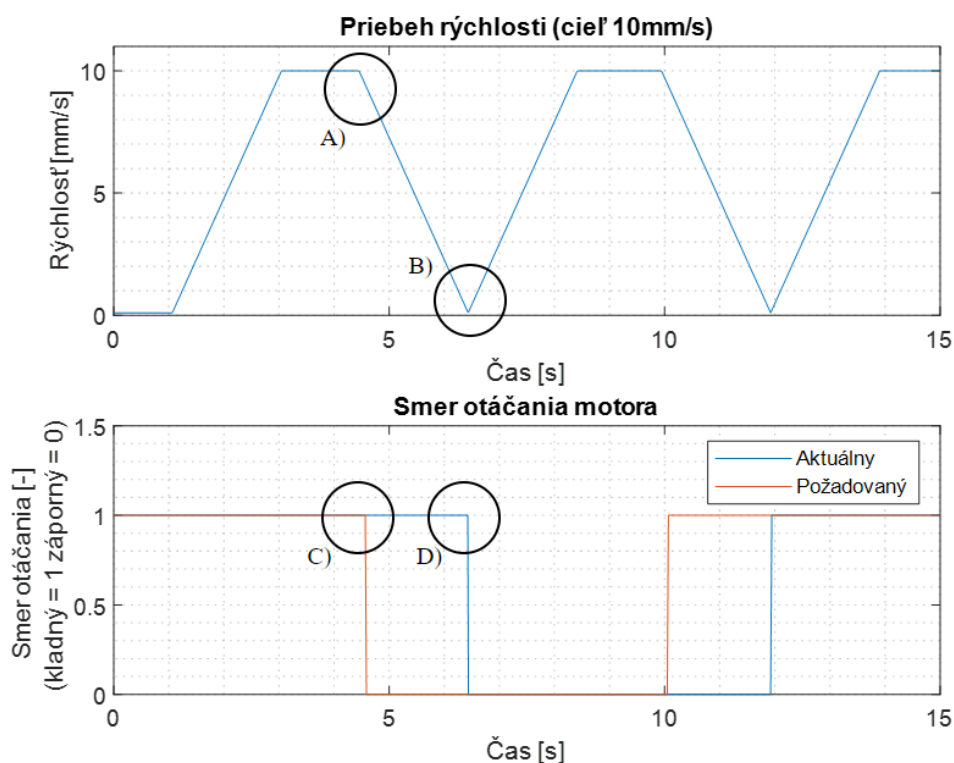


Obr. 9-2 Výsledky testovania módu automatického polohovania

9.1.2 Testovanie akcelerácie pri manuálnom polohovaní

Ako je uvedené v úvode podkapitoly 9.1 v móde manuálneho polohovania môže prichádzať ku zmene požadovaného smeru počas pohybu motora. Požadovaná rýchlosť je nastavená na 10mm/s. Pre lepšiu názornosť uvádzanej funkcionality bolo požadované zrýchlenie nastavené na 5mm/s². Grafy uvedené na Obr. 9-3 potvrdzujú funkčnosť algoritmu, ktorého úlohou je zabezpečenie decelerácie pred zmenou smeru, aby sa predišlo mechanickým rázom tak, ako je popísané v úvode podkapitoly 9.1.

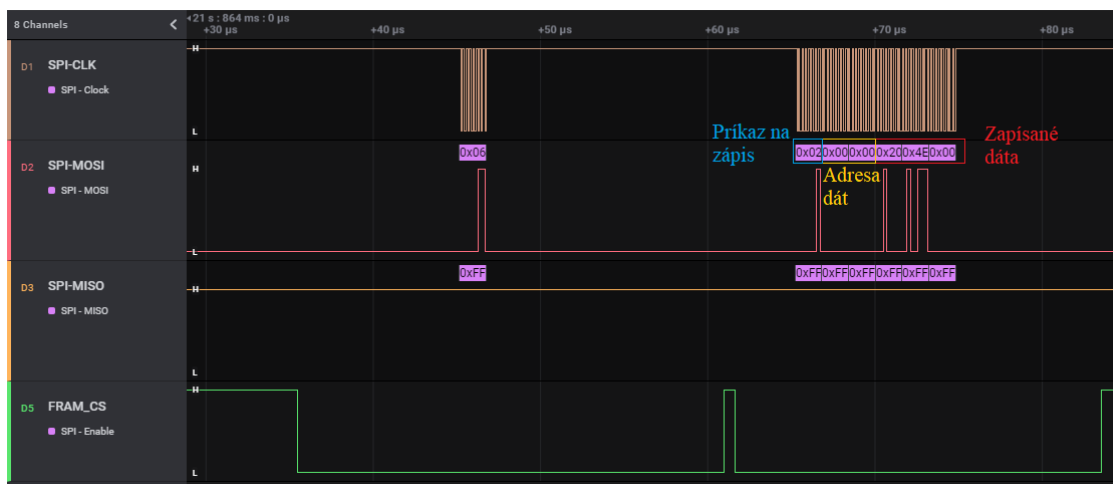
V detaile C) je možné vidieť skokovú zmenu požadovaného smeru otáčania. Keďže sa pohon pohybuje plnou rýchlosťou 10 mm/s, v detaile A) prichádza ku začiatku decelerácie. Po dosiahnutí rozbahovej rýchlosti motoru v detaile B) prichádza ku zmene smeru otáčania čoho výsledkom je skoková zmena aktuálneho smeru otáčania v detaile D) na požadovanú hodnotu. Následne motor opäť akceleruje na hodnotu rýchlosti 10mm/s.



Obr. 9-3 Výsledky testovania módu manuálneho riadenia

9.2 Testovanie zápisu do pamäte FRAM

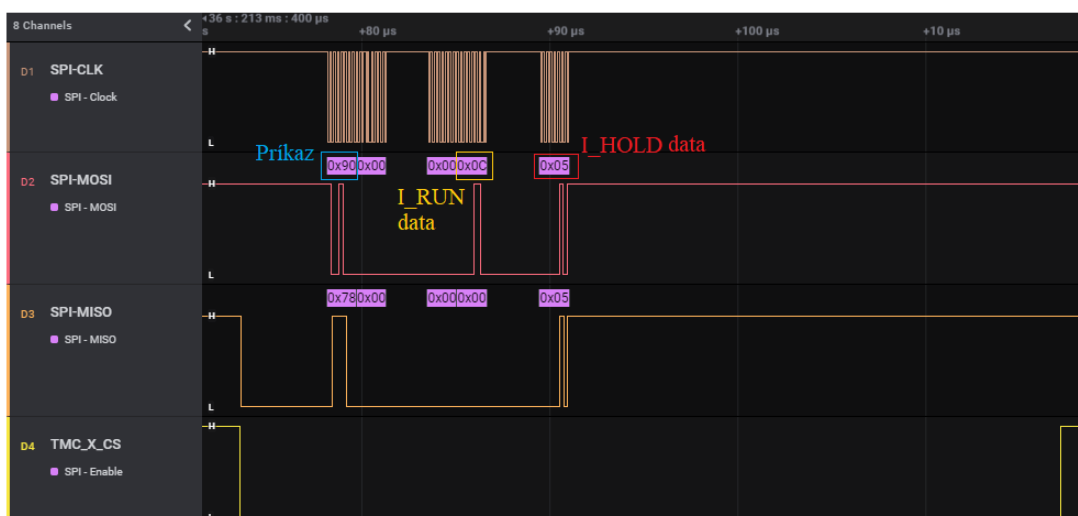
Ako je uvedené v kapitole 7.8, riadiaca jednotka disponuje pamäťou FRAM. Pri testovaní bola zadaná v užívateľskom softvari požiadavka na aktualizáciu maximálnej akcelerácie na hodnotu 20mm/s^2 . Software do riadiacej jednotky odosiela hodnotu ako $20000\text{ }\mu\text{m/s}^2$. Účelom je odstránenie nutnosti prípadnej práce s desatinnou čiarkou. Začiatková adresa, na ktorú sú dáta maximálnej akcelerácie ukladané je vo firmware riadiacej jednotky definovaná ako $0x0000$. Keďže FRAM pracuje s 8bitovými pamäťovými miestami, je potrebné hodnotu 20000 rozdeliť na jednotlivé bajty. Samotný zápis dát začína najmenej významným bajtom tak, ako uvádza Obr. 9-4.



Obr. 9-4 Aktualizácia užívateľských parametrov v pamäti FRAM

9.3 Testovanie zápisu do registrov radiča TMC2130

Obr. 9-5 uvádza testovanie funkcionality na aktualizáciu dát v registri radiča. Konkrétne sa jedná o aktualizáciu registru IHOLD_IRUN. V tomto prenose je jeho časť I_RUN aktualizovaná na hodnotu 12, ktorej hexadecimálna forma je $0x0C$. Súčasťou odosielaných dát je taktiež pôvodná hodnota časti I_HOLD aby neprišlo k prepisu dát ako uvádza podkapitola 7.7.1.



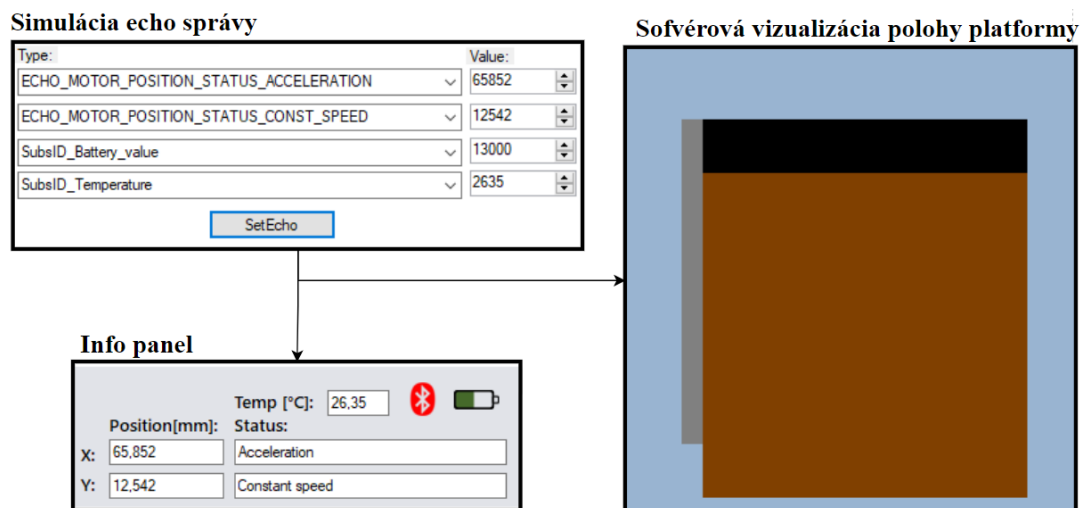
Obr. 9-5 Aktualizácia registru radiča TMC2130

9.4 Verifikácia softwarových funkcií

Za čiastočnú verifikáciu softwarových funkcií je možné považovať predchádzajúce podkapitoly, ktoré dokazujú schopnosť softwaru komunikovať s riadiacou jednotkou.

9.4.1 Simulácia prijímania správy

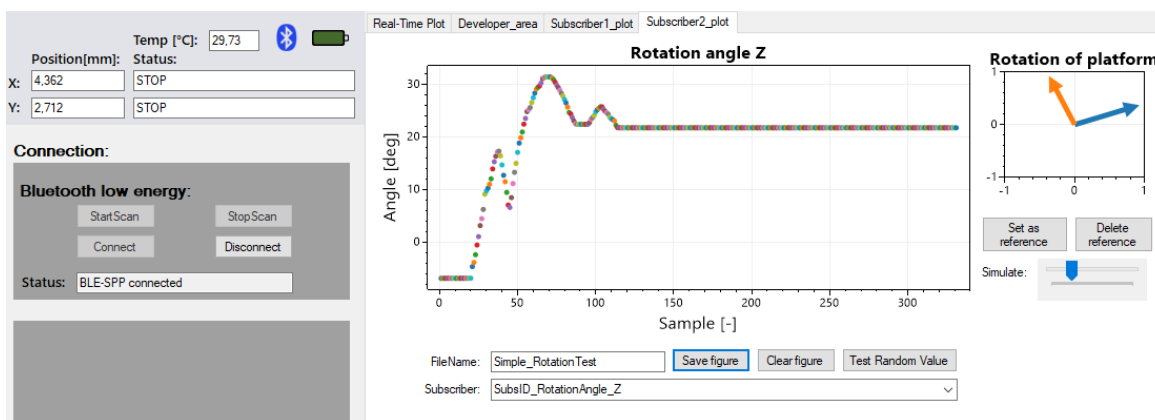
Na simuláciu echo správy je možné využiť vývojárske rozšírenie uvedené v podkapitole 8.3. Ako je vidieť na Obr. 9-6, v prípade, že nie je riadiaca jednotka platformy v dosahu PC, prichádza ku zmene farby Bluetooth ikony. Taktiež dokáže na prijatú správu reagovať ikona batérie. Pri využívaní vývojárskeho rozšírenia je potrebné uvádzať simulované echo dáta v celých číslach, keďže v tomto tvare ich software prijíma od riadiacej jednotky.



Obr. 9-6 Testovanie softwaru využitím vývojárskeho rozšírenia

9.4.2 Vykresľovanie zvolených parametrov

Pri tomto teste bol zvolený subscriber slúžiaci na vyčítavanie aktuálneho natočenia platformy. Ako je vidieť na Obr. 9-7 firmware dokáže tieto dáta na vyžiadanie odosielať, následne ich software spracuje a vykreslí do grafu, ktorý je možné uložiť. Pre lepšiu predstavu natočenia platformy je vizualizácia doplnená taktiež o natáčanie súradnicového systému. Užívateľovi je taktiež poskytnutá možnosť referencovania tohto súradnicového systému, vďaka čomu si môže určiť nulové natočenie.



Obr. 9-7 Vyčítavanie dát pomocou subscriera

10 Záver

Táto diplomová práca je súčasťou tímového projektu, ktorého cieľom je návrh univerzálneho bezdrôtového riadiaceho systému pre manipulačnú platformu napájanú batériou. Navrhovaný systém je možné rozdeliť do troch hlavných častí, ktorými sú riadiaca jednotka, jej firmware a užívateľský PC software. Návrh a výroba prototypu riadiacej jednotky sú cieľmi diplomovej práce Adama Ondryáša [39].

Hlavným cieľom tejto práce je návrh firmwaru riadiacej jednotky a užívateľského PC softwaru. Cieľom tejto práce je taktiež návrh testovacej pohonnej jednotky slúžiacej na verifikáciu funkcionality riadiaceho systému.

Kapitola 3 sa zaoberá analýzou polohovacích mechanizmov, komerčne dostupných platforiem a využívaných pohonov. Na základe nadobudnutých poznatkov bolo rozhodnuté, že kombinácia krokového motora a pohybovej skrutky je vhodným riešením testovacej pohonnej jednotky. V ďalšej časti je preto uvedené rozdelenie krokových motorov a spôsoby ich riadenia. Táto kapitola je ukončená prehľadom komerčne dostupných radičov krokových motorov, popisom typov enkodérov a koncových senzorov.

Koncepcné riešenie polohovacej platformy uvedené v kapitole 4, je doplnené o popis zvolených komponentov testovacej pohonnej jednotky. Tá sa skladá z krokového motoru s integrovanou pohybovou skrutkou a koncových senzorov s Hallovým čidlom. Keďže je jednou z požiadaviek možnosť odstránenia potreby referencovania osi po zapnutí zariadenia, bol zvolený absolútny enkodér RLS Orbis, slúžiaci na inicializačné odmeranie polohy.

Za účelom možnosti paralelného vývoja firmwaru a prototypu riadiacej jednotky, bola v rámci tejto práce navrhnutá vývojová riadiaca jednotka využívajúca STM Nucleo-F401RE. V kapitole 5 je taktiež uvedená prototypová riadiaca jednotka, ktorú navrhol Ondryáš [39]. Táto riadiaca jednotka disponuje viacerými prídavnými funkcionalitami, ktoré bolo potrebné zohľadniť pri návrhu firmwaru a softwaru.

Aby bola umožnená bezdrôtová komunikácia medzi užívateľským PC a riadiacou jednotkou polohovacej platformy, je potrebné definovať pravidlá skladby jednotlivých správ. Pri návrhu komunikačného protokolu v kapitole 6 bolo potrebné zohľadniť požiadavky na možnosť riadenia platformy v oboch smeroch súčasne a získania informácie o jej polohe, stave osí a možnosti vyčítavania zvolených dát (batéria, natočenie, nabíjací prúd...). Vďaka navrhnutému spôsobu komunikácie pomocou príkazových identifikátorov je možné jednoduché rozširovanie o nové príkazy, čo značne uľahčuje ďalší vývoj a testovanie.

Hlavným cieľ tejto práce, ktorým je návrh riadiaceho firmwaru sa zaoberá kapitola 7. Pri jeho návrhu bolo potrebné zohľadniť systémové požiadavky, ktoré v rámci svojej práce stanovil Ondryáš [39]. Vytvorený firmware pracuje na princípe stavového automatu. Vďaka tomu je možné pri ďalšom vývoji jeho jednoduché vetvenie za účelom pridávania nových funkcionalít. Navrhnutý riadiaci algoritmus prináša možnosti akcelerovaného riadenia krokových motorov a ukladania nastavených užívateľských parametrov do pamäte FRAM. Taktiež implementuje ďalšie funkcionality riadiacej jednotky, ktorými sú riadenie komponentov pomocou GPIO expandéru, odmeriavanie natočenia platformy pomocou IMU, možnosť využitia absolútnych enkodérov, spracovanie signálu od koncových senzorov, odmeriavanie teploty a komunikácia s nabíjacím modulom. Na obojsmernú komunikáciu s užívateľským PC softwarom slúži BLE modul.

Pre všetky z uvedených komponentov boli vytvorené firmwarové knižnice. Tieto knižnice obsahujú definície dátových štruktúr a funkcií slúžiacich na obsluhu jednotlivých komponentov. Pri ich návrhu bolo potrebné dbať na možnosť jednoduchšej úpravy riadiaceho algoritmu v prípade hardwarových zmien, či už riadiacej alebo pohonnej jednotky. Za dôkaz univerzálnosti navrhnutého riešenia je možné považovať prechod z vývojovej na prototypovú jednotku, keďže využívajú rôzne mikrokontroléry od spoločnosti STMicroelectrotechnic. Navrhnutý riadiaci systém bol taktiež využitý v diplomovej práci Daniela Ožvoldíka [40]. Táto aplikácia dokazuje využiteľnosť možnosti prispôbenia algoritmu zmenám komponentov pohonnej jednotky.

Riadiaci PC software uvedený v kapitole 8 poskytuje jednoduché intuitívne prostredie. Užívateľ si môže zvoliť medzi manuálnym polohovaním, alebo automatickým polohovaním, ktoré sa ďalej delí na absolútne a relatívne. Navrhnutý software taktiež umožňuje užívateľovi zmenu dynamických parametrov osí, vďaka čomu je umožnené jednoduché prispôbenie požiadavkám danej aplikácie bez nutnosti znalosti a zložitej úpravy firmwaru. Softwarové riešenie poskytuje taktiež možnosť vyčítavania a vykresľovania rôznych dát z riadiacej jednotky. Vďaka tomu užívateľ získava spätnú väzbu počas riadenia platformy. Navrhnuté riešenie ďalej obsahuje vývojárske rozšírenie, ktoré poskytuje možnosť simulácie prijatej a odosielanej správy. Software je preto možné vyvíjať a testovať nezávisle s vývojom firmwaru a riadiacej jednotky, čo prináša možnosť urýchlenia vývoja a bezpečné otestovanie nových softwarových funkcionalít pred pripojením k riadiacej jednotke.

Poslednou časťou tejto práce je testovanie funkcionalít navrhnutého firmwaru a softwaru. Za týmto účelom bola v rámci tejto práce vytvorená testovacia zostava uvedená v kapitole 9, ktorá bola pripojená ku riadiacej jednotke. Na základe uvedených výsledkov je možné považovať navrhnutý riadiaci systém za funkčný a splňujúci požiadavky, ktoré na neho boli kladené.

Celkovým výsledkom tímového projektu je funkčný prototyp bezdrôtového riadiaceho systému, ktorý môže byť aplikovaný do existujúcich konštrukčných riešení polohovacích platforiem. Pri vývoji v rámci ďalších diplomových prác je možné na tento projekt nadviazať využitím sofistikovaných funkcionalít jednotlivých komponentov riadiacej jednotky, pridávaním ďalších modulov užívateľského softwaru, prípadne systém využiť pri návrhu vlastného konštrukčného riešenia polohovacej platformy.

Bibliografia

- [1] COLLINS, Danielle. A selection guide for linear systems. *Linear motion tips* [online]. 29.6.2015 [cit. 2022-05-17]. Dostupné z: <https://www.linearmotiontips.com/selection-guide-linear-systems/>
- [2] RADEALLI, Massimo. Linear motors: Time for some straight talking. *Servotecnica* [online]. 1.4.2021 [cit. 2022-05-17]. Dostupné z: <https://www.servotecnica.com/en/resources/motion-blog-en/motori-lineari-cosa-sono-e-come-funzionano/>
- [3] RADÍČOVÁ, Tatiana. Inteligentné polohové riadenie LPMP 1: Teoretický základ. *Posterus* [online]. 23.12.2011 [cit. 2022-05-17]. Dostupné z: <http://www.posterus.sk/?p=12340>
- [4] PILine® Ultrasonic Piezomotors: Compact Drives, Fast and Self-Locking. In: *Physikinstrumente: Piezoelectric drives* [online]. [cit. 2022-05-17]. Dostupné z: <https://www.physikinstrumente.com/en/expertise/technology/piezoelectric-drives/piline-ultrasonic-motors/>
- [5] ANSELMO, Mike. Go long: The pros and cons of rack-and-pinion systems. In: *Machine Design: Linear motion* [online]. 1.2.2010 [cit. 2022-05-17]. Dostupné z: <https://www.machinedesign.com/mechanical-motion-systems/linear-motion/article/21831764/go-long-the-pros-and-cons-of-rackandpinion-systems>
- [6] Linear drives: 8 Advantages of rack and pinion. *Apex Dynamics* [online]. [cit. 2022-05-17]. Dostupné z: <https://www.apexdyna.nl/en/linear-drives-advantages-rack-pinion>
- [7] Belt Drive or Lead Screw: Belt drives and their application. In: *PBC Linear* [online]. 1.2.2020 [cit. 2022-05-18]. Dostupné z: <https://www.pbclinear.com/Blog/2020/February/Lead-Screw-or-Belt-Drives>
- [8] CASSILO, David. Belt-driven versus ball screw actuator: Which is the best choice for your application. In: *Isotechinc: Product news* [online]. 21.11.2019 [cit. 2022-05-17]. Dostupné z: <https://www.isotechinc.com/belt-driven-versus-ball-screw-actuators/>
- [9] How to Choose the Best Linear Actuator for your Application. In: *Teknic* [online]. 24.12.2020 [cit. 2022-05-18]. Dostupné z: <https://teknic.com/choose-a-linear-actuator/#ball-screws>

- [10] NOOK, Christopher. Lead Screws vs Ball Screws: It's All about the Application. In: *Helix: Lead Screws* [online]. [cit. 2022-05-18]. Dostupné z: <https://www.helixlinear.com/blog/lead-screws/lead-screws-vs-ball-screws-its-all-about-the-application/>
- [11] NOOK, Christopher. 10 Reasons Lead Screws are Better than Ball Screws. In: *Helix: Lead Screws* [online]. [cit. 2022-05-18]. Dostupné z: <https://www.helixlinear.com/blog/lead-screws/10-reasons-lead-screws-are-better-than-ball-screws/>
- [12] COLLINS, Danielle. What options are there for integrated motor and screw designs. In: *Linear motion tips* [online]. 2.7.2020 [cit. 2022-05-18]. Dostupné z: <https://www.linearmotiontips.com/what-options-are-there-for-integrated-motor-and-screw-designs/>
- [13] Motorized Linear Translation Stage Selection Guide. *Newport: Motorized Linear Stages* [online]. [cit. 2022-05-17]. Dostupné z: <https://www.newport.com/g/motorized-linear-translation-stage-selection-guide>
- [14] XY tables: XY-BSMA-200x200WJ. In: *IntelLiDrives* [online]. [cit. 2022-05-17]. Dostupné z: <https://www.intelldrives.com/XY-tables-XYZ-Systems/XY-Tables/XY-BSMA-200x200WJ>
- [15] L-731 Precision XY Stage: High Travel Accuracy and Stability. In: *Physik Instrumente: XY-stages* [online]. [cit. 2022-05-17]. Dostupné z: <https://www.physikinstrumente.com/en/products/xy-stages/l-731-precision-xy-stage-1201909/>
- [16] Motorized XY Linear Stage: ONE-XY200. In: *Newport: One-XY Integrated Linear motor XY Linear Stages* [online]. [cit. 2022-05-17]. Dostupné z: <https://www.newport.com/p/ONE-XY200>
- [17] XY positioning stage: 8MT200XY. In: *Standa: Opto-Mechanical Products* [online]. [cit. 2022-05-17]. Dostupné z: https://www.standa.lt/products/catalog/motorised_positioners?item=679&prod=xy-positioning-stage-planar
- [18] Stepper Motor And Controller Primer: Introduction. In: *Phidgets* [online]. [cit. 2022-05-17]. Dostupné z: https://www.phidgets.com/docs/Stepper_Motor_and_Controller_Primer#Introduction
- [19] What is a Stepper Motor: Types & Its Working. In: *Elprocus: Electronics projects focus* [online]. [cit. 2022-05-17]. Dostupné z: <https://www.elprocus.com/stepper-motor-types-advantages-applications/>

- [20] FIORE, Carmine. Stepper Motor Basics: Types, Uses, and Working Principles. In: *Monolithic Power Systems* [online]. [cit. 2022-05-18]. Dostupné z: <https://www.monolithicpower.com/en/stepper-motors-basics-types-uses>
- [21] Hybrid Stepper Motor: Meaning and Working of Hybrid Stepper Motor. In: *Circuit Globe* [online]. [cit. 2022-05-18]. Dostupné z: <https://circuitglobe.com/hybrid-stepper-motor.html>
- [22] Krokový motor: Hybridní krokový motor. In: *Servo-drive: FAQ* [online]. [cit. 2022-05-17]. Dostupné z: https://www.servo-drive.cz/%C4%8Dasto_pokl%C3%A1dan%C3%A9_ot%C3%A1zky_o_krokov%C3%BDch_motorech.php
- [23] Drive definitions. In: *Haydon Kerk Pittman: Learning zone* [online]. [cit. 2022-05-18]. Dostupné z: <https://www.haydonkerkpittman.com/learningzone/technicaldocuments/drive-definitions>
- [24] COLLINS, Danielle. What is a chopper drive for a stepper motor. In: *Linear motion tips* [online]. 11.8.2017 [cit. 2022-05-18]. Dostupné z: <https://www.linearmotiontips.com/what-is-a-chopper-drive-for-a-stepper-motor/>
- [25] Stepper Motor Technical Note: Microstepping Myths and Realities. In: *Faulhaber: Stepper motor tutorial* [online]. [cit. 2022-05-18]. Dostupné z: <https://www.faulhaber.com/en/support/technical-support/motors/tutorials/stepper-motor-tutorial-microstepping-myths-and-realities/>
- [26] COLLINS, Danielle. What is microstepping. In: *Linear motion tips: Microstepping basics* [online]. 21.11.2017 [cit. 2022-05-18]. Dostupné z: <https://www.linearmotiontips.com/microstepping-basics/>
- [27] Trinamic motion control: Motor & motion control basics. In: *M.C.Technology* [online]. [cit. 2022-05-18]. Dostupné z: https://www.247motioncontrol.com/media/1264/motioncontrol_motor-basics_gg.pdf
- [28] WALTER, Moritz. How accurate is microstepping really. In: *Hackaday* [online]. 29.8.2016 [cit. 2022-05-18]. Dostupné z: <https://hackaday.com/2016/08/29/how-accurate-is-microstepping-really/>
- [29] COLLINS, Danielle. What is current decay (aka recirculating current) in a stepper drive. In: *Motion control tips* [online]. 27.12.2018 [cit. 2022-05-18]. Dostupné z: <https://www.motioncontroltips.com/what-is-current-decay-in-a-stepper-drive/>

- [30] Optimized Current Control. In: *ADI Trinamic* [online]. [cit. 2022-05-18]. Dostupné z: <https://www.trinamic.com/technology/motor-control-technology/chopper-modes/>
- [31] DRV8833: dual H-bridge motor driver with current regulation. In: *Texas Instruments: Stepper motor drivers* [online]. [cit. 2022-05-18]. Dostupné z: <https://www.ti.com/document-viewer/DRV8833/datasheet>
- [32] L6208 DMOS driver for bipolar stepper motor: Datasheet - production data. In: *STMicroelectronics: Stepper Motor Drivers* [online]. [cit. 2022-05-18]. Dostupné z: <https://www.st.com/resource/en/datasheet/l6208.pdf>
- [33] TMC2130: Datasheet. In: *ADI Trinamic: Integrated Circuits* [online]. [cit. 2022-05-18]. Dostupné z: https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC2130_datasheet_rev1.14.pdf
- [34] EDWARDS, Ed. Types of Encoders: Rotary, Linear, Position and Optical Encoder Types. In: *Thomas: Automation electronics* [online]. [cit. 2022-05-18]. Dostupné z: <https://www.thomasnet.com/articles/automation-electronics/types-of-encoders-a-thomasnet-buying-guide/>
- [35] Orbis: Rotary Absolute Magnetic Encoder Module. In: *RLS: Absolute rotary encoders* [online]. [cit. 2022-05-18]. Dostupné z: <https://www.rls.si/eng/orbis-true-absolute-rotary-encoder>
- [36] Types of Endstops. In: *Toglefritz's Lair: 3D printing* [online]. 27.7.2015 [cit. 2022-05-18]. Dostupné z: <https://toglefritz.com/types-of-endstops/>
- [37] Krokový motor s integrovaným trapézem. In: *NA3D: Krokové motory* [online]. [cit. 2022-05-18]. Dostupné z: <https://www.na3d.cz/p/2395/krokovy-motor-056-nm-s-integrovanym-trapezem-t8x8-a-matici>
- [38] STM32_Trinamic_Stepper: TMC2130. In: *Github: alambe94* [online]. [cit. 2022-05-19]. Dostupné z: https://github.com/alambe94/STM32_Trinamic_Stepper/tree/master/TMC2130
- [39] ONDRYÁŠ, Adam. *Návrh a realizace elektroniky pro polohovací platformu* [online]. Brno, 2022 [cit. 2022-05-18]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/139929>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Zdeněk Hadaš.
- [40] OŽVOLDÍK, Daniel. *Návrh a analýza konstrukce polohovací jednotky* [online]. Brno, 2022 [cit. 2022-05-19]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/140176>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav výrobních strojů, systémů a robotiky. Vedoucí práce Rostislav Huzlík.

Zoznam obrázkov

Obr. 2-1 Navrhované bezdrôtové riešenie	11
Obr. 2-2 Schéma cieľov práce	12
Obr. 3-1 Lineárny motor [1]	14
Obr. 3-2 Časti piezomotoru - upravené z [4]	14
Obr. 3-3 Mechanizmus kombinujúci pastorok a ozubený hrebeň [6]	15
Obr. 3-4 Mechanizmus využívajúci ozubený remeň [1]	15
Obr. 3-5 Rotačný motor s integrovanou pohybovou skrutkou [12]	16
Obr. 3-6 Platforma XY-BSMA [14]	18
Obr. 3-7 L-731 Precision XY Stage [15]	18
Obr. 3-8 Platforma ONE-XY200 [16]	19
Obr. 3-9 Platforma 8MT200XY [17]	19
Obr. 3-10 Schéma hybridné krokového motora [22]	21
Obr. 3-11 Unipolárny krokový motor [20]	22
Obr. 3-12 Bipolárny krokový motor [20]	22
Obr. 3-13 Schéma odmeriavania prúdu v cievke motora, upravené z [27]	23
Obr. 3-14 Porovnanie vplyvu mikrokrokovania [27]	24
Obr. 3-15 Porovnanie typov zániku prúdu [29]	25
Obr. 3-16 Integrovaný obvod DRV8833 [31]	26
Obr. 3-17 STMicroelectronics L6208 [32]	27
Obr. 3-18 Schéma vnútornej štruktúry radiča TMC2130 [33]	28
Obr. 4-1 Schéma koncepčného riešenia polohovacej platformy	30
Obr. 4-2 Sústava telies	31
Obr. 4-3 Krokový motor s integrovanou pohybovou skrutkou [37]	33
Obr. 4-4 Absolútny enkodér RLS Orbis [35]	34
Obr. 4-5 Koncový senzor s Hallovým čidlom [39]	34
Obr. 5-1 Vývojová riadiaca jednotka	35
Obr. 5-2 Finálne riešenie riadiacej jednotky (pohľad zhora) [39]	36
Obr. 5-3 Finálne riešenie riadiacej jednotky (pohľad zdola) [39]	36
Obr. 6-1 Typy správ v bezdrôtovej komunikácii	37
Obr. 6-2 Zloženie príkazovej správy	38
Obr. 6-3 Základný princíp vyhodnocovania identifikátorov príkazov	38
Obr. 6-4 Zloženie echo správy	39
Obr. 6-5 Základný princíp vyhodnocovania echo identifikátorov	39
Obr. 7-1 Obecná schéma vytváraných knižníc	41
Obr. 7-2 Zloženie dátovej štruktúry na obsluhu komponentu	42
Obr. 7-3 Príklad zloženia dátovej štruktúry na obsluhu TMC2130	43
Obr. 7-4 Schéma štruktúry riadiaceho algoritmu	45
Obr. 7-5 Príklad deklarácie premenných a inicializácie firmwaru	46
Obr. 7-6 Schéma hlavného stavového automatu	47
Obr. 7-7 Schéma procesov na začiatku cyklu stavového automatu	48
Obr. 7-8 Koniec cyklu stavového automatu	49
Obr. 7-9 Princíp vytvorenia časti echo správy	50
Obr. 7-10 Schéma riadiaceho systému pre krokový motor	51
Obr. 7-11 Nastavovanie registrov radiča TMC2130	52

Obr. 7-12 Kombinácia dát pri zápise do registru radiča TMC2130.....	53
Obr. 7-13 Porovnanie výhod využitia funkcií knižnice.....	53
Obr. 7-14 Schéma princípu nastavenia registrov TMC2130 využitím funkcií	54
Obr. 7-15 Schéma komunikácie enkodéru RLS ORBIS	56
Obr. 7-16 Stavový automat na riadenie krokového motora.....	57
Obr. 7-17 Diagram obsluhy pamäte FRAM	59
Obr. 7-18 Porovnanie využitia GPIO expandéru a digitálneho pinu mikrokontroléru	60
Obr. 8-1 Rozloženie základného užívateľského prostredia	64
Obr. 8-2 Možnosti dynamického riadiaceho panelu.....	65
Obr. 8-3 Statický riadiaci panel	65
Obr. 8-4 Vizualizácia vývojárskeho rozšírenia	66
Obr. 8-5 Rozdelenie vytvorených softwarových tried.....	67
Obr. 8-6 Schéma procesu odoslania príkazovej správy	69
Obr. 8-7 Schéma procesu spracovania echo správy	69
Obr. 9-1 Testovacia zostava s riadiacou jednotkou	70
Obr. 9-2 Výsledky testovania módu automatického polohovania.....	71
Obr. 9-3 Výsledky testovania módu manuálneho riadenia.....	72
Obr. 9-4 Aktualizácia užívateľských parametrov v pamäti FRAM	73
Obr. 9-5 Aktualizácia registru radiča TMC2130	73
Obr. 9-6 Testovanie softwaru využitím vývojárskeho rozšírenia.....	74
Obr. 9-7 Vyčítavanie dát pomocou subscribera	74

Zoznam tabuliek

Tab. 3-1 Porovnanie pohonov polohovacích platforiem [13].....	17
Tab. 4-1 Parametre zvoleného krokového motoru [37].....	33
Tab. 7-1 Zoznam vytvorených knižníc na obsluhu komponentov	44
Tab. 7-2 Význam indikátorov statusu BLE modulu.....	50
Tab. 7-3 Podmienky zmien stavov automatu krokového motora	57
Tab. 7-4 Signalizačné sekvencie RGB LED krúžku	62
Tab. 8-1 Základné metódy triedy BLE_comm.....	68
Tab. 8-2 Udalosti definované v triede BLE_comm.....	68

Zoznam skratiek

ADC	Analog-to-digital converter
ARR	AutoReload register
BLE	Bluetooth Low energy
CS	Chip select
DIR	Direction
FRAM	Ferroelectric RAM
GPIO	General-purpose input/output
ID	Identifikátor
IMU	Inertial measurement unit
IO	Input/output
I2C	Inter-Integrated Circuit
KM	Krokový motor
LED	Light-Emitting Diode
MISO	Master in slave out
MOSI	Master out slave in
PSC	Prescaler
PWM	Pulse width modulation
RGB	Red Green Blue
SPI	Serial Peripheral Interface
TIM	Timer peripheral
UART	Universal asynchronous receiver-transmitter

Zoznam príloh

Názov zložky	Názov podzložky	Obsah
ControlSoftware	/ControlSoftwareApp	Užívateľský PC software
Firmware_L476	/Core/Inc	Zložka hlavičkových súborov firmwaru
	/Core/Src	Zložka zdrojových súborov firmwaru
	/STM32L476.ioc	STM CUBE MX projekt